

TST

ANALYSIS AND SYNTHESIS

OF

TIME-SHARED COMPUTER SYSTEMS

BY

ARNE NILSSON

1976

COMMUNICATION SYSTEMS
LUND INSTITUTE OF TECHNOLOGY

TELETRAFIKSYSTEM

TEKNISKA HÖGSKOLAN I LUND

Box 725 - 220 07 LUND - Sweden

ANALYSIS AND SYNTHESIS
OF
TIME-SHARED COMPUTER SYSTEMS

AKADEMISK AVHANDLING

som för avläggande av teknisk doktorsexamen vid tekniska fakulteten vid universitetet i Lund kommer att offentligen försvaras i sal E:A, Lunds tekniska högskola, fredagen 1976-03-26 kl 10.15.

av

Arne Nilsson
civ ing, Ssk



Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41553215_0001

Acknowledgements

I would like to express my appreciation to Professor Bengt Wallström for his continued encouragement and his friendship during the course of this research. I am also deeply indebted to Dr. Leonard Kleinrock at the Department of Computer Science, UCLA for his constant encouragement, his numerous suggestions and his friendship during the year I spent at UCLA.

A debt of gratitude is also owed to the personnel of the Telecommunication Systems Department at Lund Institute of Technology for providing the stimulating atmosphere in which part of this research was performed. Special thanks go to Anna Odin and Penny Kenton for their patient assistance in typing and preparation of this dissertation.

Without the support and encouragement of my wife, especially her ability to transform my English into standard English, none of this would have been possible.

I am also grateful to the Ericsson Telephone Company for their support during my year at UCLA. Finally, it is my pleasure to acknowledge the support given by the Swedish National Board for Technical Development, and the Swedish Telecommunications Administration.

CONTENTS

		<u>Page</u>
Chapter 1.	Introduction	1:1
Chapter 2.	Definitions and Preliminaries	2:1
2.1	The Queueing Structure	2:4
2.2	The M/G/1 Queue	2:8
2.3	Distribution of Attained Service	2:11
Chapter 3.	Scheduling Algorithms, Some Known Results	3:1
3.1	The Batch Processing Algorithm	3:1
3.2	The Round Robin Scheduling Algorithm	3:3
3.3	The Last-Come-First-Served-Preemptive-Resume Scheduling Algorithm	3:5
3.4	The Foreground Background Scheduling Algorithm	3:7
3.5	Some Numerical Examples	3:12

Chapter 4.	The Multilevel (ML) Scheduling Algorithm	4:1
Chapter 5.	Selfish Scheduling Algorithms	5:1
Chapter 6.	Bounds, Inequalities and a Conservation Law	6:1
Chapter 7.	The Variance of Conditional Response Time	7:1
7.1	FCFS-Scheduling	7:2
7.2	FB-Scheduling	7:3
7.3	LCFS+PR-Scheduling	7:4
7.4	RR-Scheduling	7:7
7.5	Did We Get a Better Measure of System Performance?	7:28
Chapter 8.	Optimal Scheduling Algorithms	8:1
8.1	The Optimization Problem	8:1
8.2	Simple Examples	8:5
8.3	The Problem in Reverse (i.e. "Stacking" the Deck)	8:8

Chapter 9.	Numerical Results for the Optimal Theory	9:1
Chapter 10.	Attempts to Optimize Multi-Level System	10:1
Chapter 11.	Conclusions and Suggestions for Future Research	11:1

1. Introduction

Most scientists of today use a computer now and then. It is probably true that the enormous technological progress made during the last 20 years could not have been achieved without the help of those formidable calculating machines. Had it been possible, for instance, for mankind to make a giant step on the moon, or classify the RNA and DNA compounds without the help of a computer? Quite probably mankind could have managed but in a far greater length of time. Nowadays, we are so used to the power and speed of the computers that a machine that can perform one million additions in less than one second is just what we expect. We also expect the computer to deliver rapid and reliable responses as soon as we have put our problem in a suitable form for the machine. This is what we expect, but are our expectations always fulfilled? Most users of a computer know that this is often not the case. This author, at least, has frequently experienced the frustrating situation where he has had to wait almost indefinitely before getting his program back from the computing facility.

To be fair, we do not always expect a rapid response from the computer. If, for instance, we submit a very

Large program to the computer we are quite willing to accept some delay before getting any result.

As a matter of fact, we gladly accept some delay because that implies, we believe, that the program is running smoothly.

On the other hand, if we submit a short job we expect a rapid response from the computer, otherwise we get annoyed and start believing that the program is hung up in an infinite loop. Clearly we recognize different needs for different users of a computing facility. We choose, however, to formulate the needs in the following, maybe somewhat pragmatic, way;

"Short jobs should be processed quickly by the computer and long jobs might take more time".

This is the principle behind the time sharing systems (TS - systems) that were constructed during the mid-sixties.

Already in the late 1950's one realized that the mismatch between the computer and the user regarding their speed, implied that neither of the two were used effectively. In the 1950's and, sad to say, even nowadays, it was not uncommon to see a programmer sitting at the computer and interacting with it on a one-for-one basis for hours at a time. The user would spend a period thinking and generating a request for computation, then wait

while that computation was carried out by the machine. This mode of operation was obviously highly efficient from the user's point of view, on the other hand, the machine's capacity was certainly not fully employed. In the late 1950's and early 1960's the user was forced away from the machine and had to submit his requests to an operator. The operator would then later return the result of the computation. This operational method kept the machine quite busy and was therefore efficient in terms of machine utilization. It was, however, most inefficient from the user's point of view, since the user often had to wait for hours (sometimes days) before he got back the result of the computation.

The introduction of time sharing systems meant that many users simultaneously could get access to the computing facility and thus share the computer's capacity. The main idea of time sharing is that significantly more time is spent by the user thinking out new problems than in the computing phase in most cases and, therefore, computation may take place on one user's request while most of the other users are in their thinking phase. If properly constructed and handled, a time sharing system can be a most effective way of distributing computer power among users. During the last decade, time sharing systems have been analysed by many researchers:

(ESTR 67, KLEI 64, KLEI 67, KLEI 70a, KLEI 75b, MCKI 69, SCHR 67).

In most studies, however, the method used is to analyse a given time sharing configuration trying to find suitable descriptors for the system's capacity. The important and interesting question regarding how to build an optimal time sharing configuration has almost been neglected. Furthermore, the analysis of given computer systems has been concentrated on the finding of expressions for average values, - the average response time, average throughput, etc. In this dissertation we shall review some of the analytical results obtained, as well as providing some new results. Furthermore, a method for optimizing (synthesis) time sharing systems will also be presented .

In Chapter 2 we introduce the computer system to be analysed and synthesized as well as the notations that will be used in the sequel. A general result for the average number of jobs present in the computer system will also be given.

In Chapter 3 analytical results are presented for some well-known configurations. These results have all been presented elsewhere.

In Chapter 4 the multilevel system is described and analysed. There we obtain some new analytical results especially for the scheduling algorithms RR and LCFS-PR.

In Chapter 5 the concept of the selfish scheduling algorithms (KLEI 70a) is presented as well as some analytical results for these algorithms.

In Chapter 6 we present tight upper and lower bounds for the response time as well as a conservation law (KLEI 71a).

In Chapter 7 we present results for the variance of conditional response time. In this chapter our interest, is mainly focused on the RR scheduling algorithm and we manage to find a lot of new results for this scheduling algorithm.

In Chapter 8 a theory for the optimization of time sharing systems is presented. We also give examples that illustrate this theory.

In Chapter 9 we give numerical results for the optimization procedure that we introduced in chapter 8.

In Chapter 10 the optimization of multilevel systems is treated and we give suggestions how to tackle this important optimization problem.

In Chapter 11 we give some concluding remarks and suggest topics of future research interests.

2. Definitions and Preliminaries.

When we want to study a computer system we are faced with the problem of trying to sort out the important parts of the system for performing a pertinent analysis and/or synthesis. Here we are especially interested in time shared (TS) systems and therefore we are concerned with a collection of remote terminals from which users may gain simultaneous access to a collection of computing resources commonly referred to as a computer system. Figure 2:1 shows a typical configuration for this computer system.

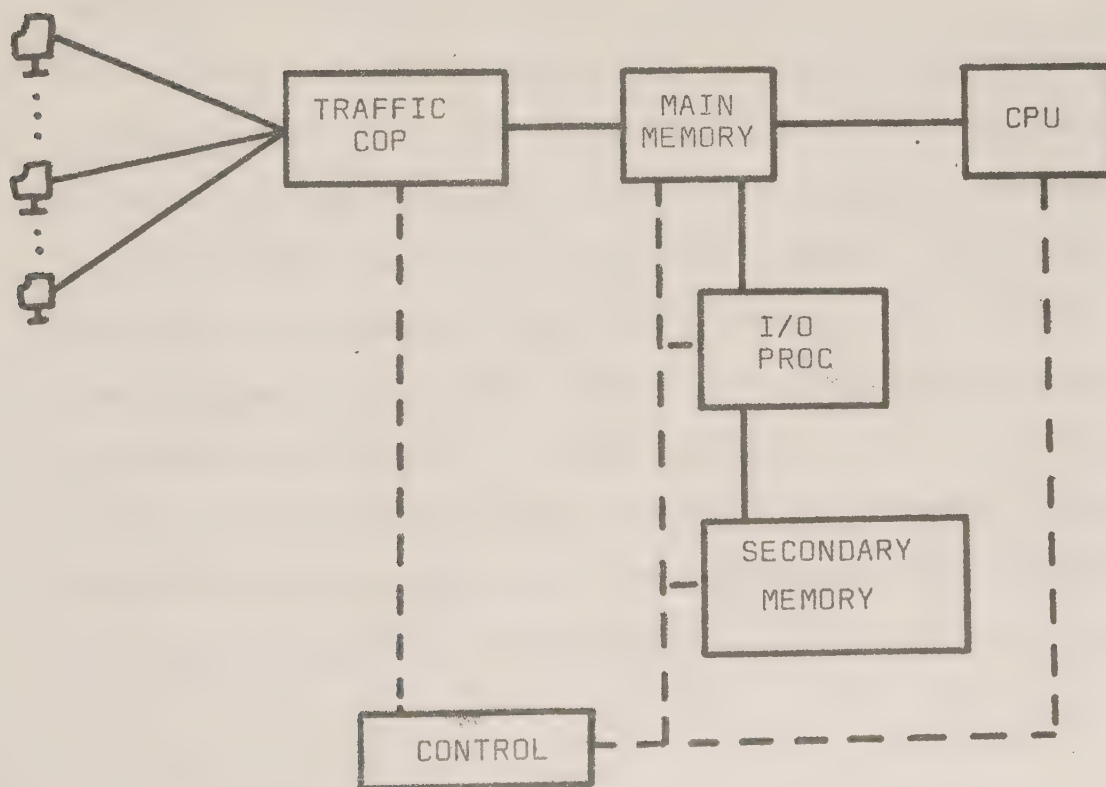


Fig. 2:1 Computer System
 (denotes data flow, denotes control)

We see that the resources include the terminals themselves, the communication lines connecting the terminals to the computer system, the traffic handler, a so called CDP, controlling the flow of traffic between the users and the computer system. The main memory and secondary memory devices e.g. drums, discs, etc., are also typical resources in the computer system as are the various types of I/O media which may be present. Lastly, the most important resource of them all: the Central Processing Unit (CPU) whose processing capacity most certainly will be a dominant factor if we try to determine the processing capacity of the computer system. Clearly we will also need a control function which governs the activity and assignment of resources in the system.

The control function is basically a supervisory program often referred to as the operating system. Quite obviously a number of conflicts may arise in this system when many user programs compete for resources, and it is my intention to study a subclass of these conflicts in this thesis, namely those arising when users (jobs, programs) compete for access to the CPU. In 1964 Kleinrock (KLEI 64) presented the first model for a time-shared computer system and the analysis performed by Kleinrock and his successors in this field made me interested in this area of research.

2.1 The Queueing Structure

Kleinrock studied time-shared systems as a class of feedback queueing systems and the same approach will be adopted by me.

These systems consist of a single resource, the CPU, the server, and a system of queues in front of the server, see figure 2:2

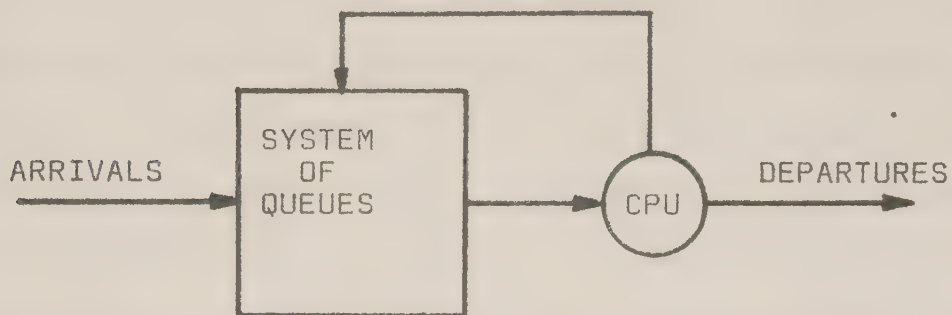


Figure 2:2

Feedback queueing system

The server has the option of sending back a job to the queues when this job has been given some amount of service. Whenever the server becomes idle a customer is brought from the queues into the server. The selection determining which customer to pick is made by the scheduling discipline.

The scheduling discipline plays an important part in the analysis of time sharing systems, as we will soon see. Therefore, let us now define the class of scheduling disciplines that we will allow. Any disciplines that do not use information about an individual customer's service requirement or some measure of that service need, will be acceptable to us. This means that we exclude such disciplines as SJF (Shortest Job First), LJF (Longest Job First), etc., since these disciplines use information about the individual customer's service requirement. This might seem contradictory to the goal formulated in the introduction, since there we stated that we would like to provide rapid service for short jobs. The schedule (discipline) should then be able to measure exactly the individual length of each job submitted to the computer structure. Such a task is obviously not feasible since the schedule has no prior knowledge of the run time needed for the jobs. Why then don't we allow the individual jobs to state their needs? Eg: upon entry to the computing facility. If the needs stated were accurate this could very well be an excellent method. But do we really expect those stated needs to be reliable? Most computer systems require that the user specifies an upper limit for the run time, but no system manager would believe a programmer who claims that his job will need exactly 10.2 CPU-seconds.

Instead, the position taken in most time sharing configurations is to let the jobs themselves prove that they are short and therefore should be given immediate attention. This operation is accomplished through the following strategy:

A new job is given a quantum of service as soon as possible. If the job is finished during that quantum it will depart from the system and so, obviously, it was a short job (unless the quantum size is foolishly chosen). If the job needs more service, it will not finish during the first quantum, it will be sent back to the system of queues and after some appropriate time, once again be given access to the server, where it will receive a second quantum of service. Should the job finish during this second quantum it leaves the system (the job was almost short), if not it will again be sent back to the queues and it will have to wait for further quanta. Let us introduce the notation

q_n = quantum of service that a job gets when it arrives at the server for the n :th time

Later on when we analyse and optimize time sharing systems we will let the q_n 's be infinitesimal, i.e. $q_n \rightarrow 0$.

This assumption will permit us to analyse a large variety of different scheduling algorithms, and experience gained from measurements and simulations tells us that this assumption does not carry us far away from the real world.

Whenever a job is transferred from the queues to the server, or from the server to the queues, the program has to be moved from the queueing media into the core, and vice versa. The time it takes to perform this transfer is usually called swap-time. In the typical jargon of the computer aficionados: Programs are "swapped" in and out of the CPU. Swap-time has in most studies of time sharing systems been neglected and we choose to do the same in this work.

In the introduction we said that one objective for a time sharing system would be to provide rapid response to short jobs. How do we then measure the quality of a time sharing system? In fact, what kind of parameters would we like to know or even like to control? The most used parameter has so far been the conditional mean response time (and/or the conditional mean waiting time), defined through

$T(x) \triangleq$ average response time for a customer requiring x seconds of processing

and

$W(x) \triangleq T(x) - x$

Often $W(x)$ is called the mean wasted time since it is obviously equal to the time a customer has to spend in the queues.

2.2 The M/G/1 Queue

We will analyse our feedback queueing system under the assumption that it is an M/G/1 queue.

This means that the arrival process is Poisson, with mean arrival intensity denoted λ , and that the service time distribution is arbitrary (General). We assume that the i^{th} arriving customer to the system has a service requirement x_i that is a stochastic variable, where

$$P(x_i \leq x) = B(x)$$

independent of i . If we don't want to specify that it is the service requirement for the i^{th} customer we will not use any subscript, i.e.,

$$x_i \rightarrow \tilde{x}$$

and naturally

$$P(\tilde{x} \leq x) = B(x)$$

The n^{th} moment of service time distribution will be denoted by $\overline{x^n}$ and is computed as

$$\overline{x^n} = \int_0^{\infty} x^n dB(x)$$

or, if we prefer,

$$\overline{x^n} = \int_0^{\infty} x^n b(x) dx$$

where $b(x)$ is the density function of the service time. Often we will also have the opportunity to work with the Laplace transform $B^*(s)$ of the service time distribution.

$$B^*(s) = E\{e^{-s\tilde{x}}\} = \int_0^{\infty} e^{-sx} b(x) dx$$

The M/G/1 queueing system has been extensively studied in the past, e.g: (TAKA 62, RIOR 62, COHE 69, KLEI 75a), and the reader looking for a more detailed treatment is referred to one of those textbooks. (The reader who chooses to study (KLEI 75a) will note that the notations used by Kleinrock are the same as used here). When we study an M/G/1 queue and want to obtain numerical results, we have to specify the service time distribution. In this study we will often choose $B(x)$ as an Erlang- r distribution, and/or an H_2 -distribution, i.e.

$$b(x) = \begin{cases} \frac{(r\mu x)^{r-1}}{(r-1)!} e^{-r\mu x} & \text{Erlang-}r \\ & r = 1, 2, \dots \\ \alpha\mu_1 e^{-\mu_1 x} + (1-\alpha)\mu_2 e^{-\mu_2 x} & H_2 \quad 0 \leq \alpha \leq 1 \end{cases}$$

The reason for this particular choice of service time distribution is that we will then get the opportunity to model a fairly large class of service time distributions. If we define the coefficient of variation C_b^2 for the service times as

$$C_b^2 = \frac{\overline{x^2}}{\overline{x}^2} - 1$$

we will find that for an Erlang- r distribution

$$C_b^2 \leq 1$$

and for an H_2 distribution

$$C_b^2 \geq 1$$

We can see that the Erlang-1 distribution is the same as the exponential distribution, thus we have also incorporated the classical M/M/1 system among the systems that we are especially interested in.

2.3 Distribution of Attained Service

In all of the feedback queueing systems to be studied below, it is clear that we will, at any point of time, have jobs scattered through the system, each job having received various amounts of useful service, i.e. attained service. The distribution of attained service for those customers still in the system was found in (KLEI 67), and recently a simpler derivation was presented in (ODON 74). This distribution will not be used too often in the sequel - in fact, the only place where it does appear is in equation (6.18) chapter 6. We do believe, however, that it would be an oversight not to present the distribution, and therefore let us define

$n(x)$ Δ average density of customers still in the system who have so far received x seconds of service.

The interpretation of this density function is given in the usual sense of density functions, namely,

$$\int_{x_1}^{x_2} n(x) dx = \text{average number of customers still in the system who have so far received between } x_1 \text{ and } x_2 \text{ seconds of service.}$$

The density $n(x)$ can be expressed as

$$n(x) = \lambda(1-B(x)) \frac{dT}{dx}$$

This result was first found by Kleinrock (KLEI 67), who used a finite quantum discussion in order to obtain his result.

3. Scheduling Algorithms, Some Known Results.

In this chapter we are going to present results for the conditional mean response time $T(x)$ and/or the conditional mean waiting time $W(x)$ for different scheduling algorithms. The results have been published elsewhere (ESTR 67, KLEI 64, KLEI 67, KLEI 71, KLEI 72a, KLEI 75b, MCKI 69, SCHR 67), but we feel that no harm is done if we repeat them here. Since the best presentation of the results, as known to this author appears in the recent book by Kleinrock (KLEI 75b), I have here chosen to follow his line of presentation.

3.1 The Batch Processing Algorithm.

The simplest algorithm we can consider is the case of batch processing, and we choose a first-come-first-served discipline. The cyclic queue model then loses its cycliness, and can be depicted as in figure 3:1.



Fig 3:1

Figure 3:1 implies that we have assumed all quanta to be infinite in duration, thereby giving us an M/G/1 FCFS

system. Another way to look upon this system is to assume infinitesimal quanta, where the customer who is ejected from the server due to the termination of quanta is immediately fed back into the server. In that case we can use fig 3:2 as a pedagogical description of the system.



Fig 3:2

For an FCFS system we know that the average waiting time is given by the Pollaczek-Khinchine formula.

$$W = \frac{\overline{\lambda x^2}}{2(1-\rho)} \quad (3.1)$$

and that the mean waiting time is independent of the service required. We can therefore conclude that

$$T(x) = \frac{\overline{\lambda x^2}}{2(1-\rho)} + x \quad (3.2)$$

or - if we want to use $W(x)$ - that

$$W(x) = \frac{\overline{\lambda x^2}}{2(1-\rho)} \quad (3.3)$$

Batchprocessing, we note, must be a very poor candidate for a time-sharing algorithm, since it does not at all discriminate with respect to service time requirement.

3.2 The Round-Robin Scheduling Algorithm.

Probably the most well-known and widely used scheduling algorithm for time-shared computer systems is the round-robin (RR) algorithm. The name round-robin is perhaps somewhat puzzling and I will here take the opportunity to explain the meaning in somewhat more detail. From the beginning a round-robin is a child's toy consisting of a circle on whose circumference preferably wooden toy-robins are sitting. As the circle turns the robins pass a bowl of some nourishment which they nibble at in turn. The same procedure is employed when scheduling jobs in a computer system that operates under the RR scheduling discipline.

The discipline can be depicted as in figure 3:3.



Fig 3:3

A new customer joins the single queue and works his way up to the head of the queue in a FCFS fashion,

where he then receives a quantum of service. When that quantum is finished, and if he needs further service, he is sent back to the tail of the queue. In the zero-quantum case it is clear that the customers have to cycle an infinite number of times in the system until their attained service is equal to their required service, at which time they depart from the system. Being popular and easily implemented, great attention has been given to the analysis of this scheduling discipline (COFF 70, KLEI 64, KLEI 75b). Sakata (SAKA 71) was perhaps the first to study the M/G/1 case, and he found that the solution for the average response time was independent of the form of the service time distribution (dependent only on the mean service time $\bar{x}$). It was found that

$$T(x) = \frac{x}{1-\rho} \quad (3.4)$$

or that

$$W(x) = \frac{x\rho}{1-\rho} \quad (3.5)$$

What a delightful property the RR-algorithm possesses! It is extremely fair, its discrimination is linear. This implies that a job twice as long as another one will spend, on average, twice as long a time in the system.

The response time is, furthermore, independent of the service time distribution $B(x)$, and depends only upon the mean value of the service time through the load on the system $\rho = \lambda \bar{x}$.

3.3 The Last-Come-First-Served-Preemptive-Resume Scheduling Algorithm.

A scheduling algorithm that deserves some attention because of its interesting properties and simple implementation, is the last-come-first-served-preemptive-resume (LCFS-PR) scheduling discipline. The structure of the system is shown in figure 3:4. As usual we limit ourselves to the zero-quantum case.

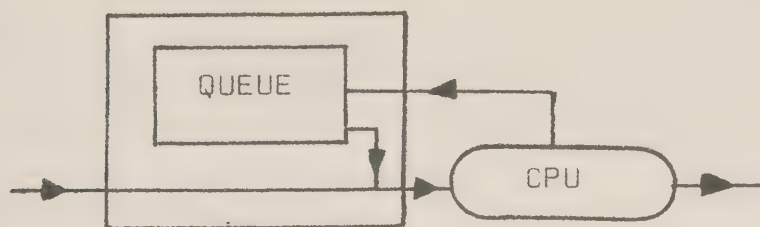


Fig 3:4

A newly arriving customer will preempt the customer in the CPU, if any, and that customer will be sent to the head of the queue. When there is a departure from the

system, the job - if any - at the head of the queue will be given service, thus implementing the LCFS rule. The average conditional response time is easily solved, since it must consist of x (the tagged job's service requirement) plus a term equaling the average time spent in the queue by the tagged job. This term is calculated as

$$\underbrace{\lambda x}_{\substack{\text{expected} \\ \text{number of} \\ \text{times the} \\ \text{job is} \\ \text{preempted}}} \cdot \underbrace{\frac{\bar{x}}{1-\rho}}_{\substack{\text{average time/preemption} \\ \text{the job spends in} \\ \text{the queue}}}$$

Hence

$$T(x) = x + \frac{\lambda x \bar{x}}{1-\rho} \quad (3.6)$$

which is, simplified,

$$T(x) = \frac{x}{1-\rho} \quad (3.7)$$

or

$$W(x) = \frac{x\rho}{1-\rho} \quad (3.8)$$

This is, however, exactly the same response time as for the RR-scheduling algorithm. Both results have been obtained for an M/G/1 system, which, as pointed out in (KLEI 75b), indicates that the conditional mean response time is not a good indicator of system performance. In chapter 7 I will use the variance of the conditional time as a complementary indicator and it will provide a better discrimination among the disciplines.

3.4 The Foreground Background Scheduling Algorithm.

The Foreground Background (FB) discipline is a generalization of the classical queueing system, that consists of a foreground queue and a background queue (COFF 66), figure 3:5

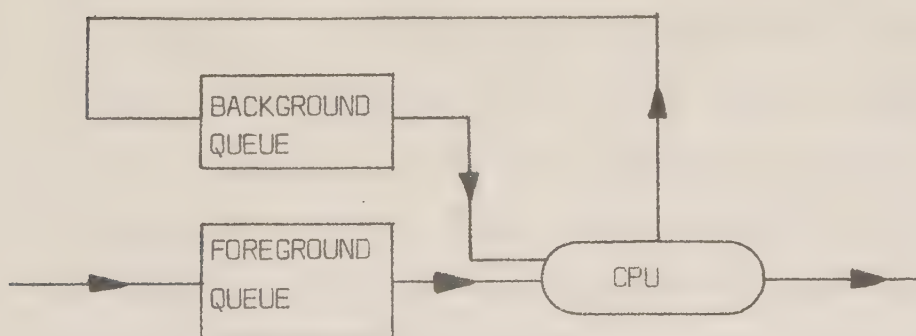


Fig 3:5

The classical queueing systems work with finite quanta.

A new arrival will enter the tail of the foreground queue and work his way up to the server in a FCFS fashion. Whenever there are customers in the foreground queue the server's interest is directed entirely towards this queue. When a customer gets to the server for the first time he is given a quantum of service; when that quantum expires he is sent to the tail of the background queue, if he needs more service, otherwise the customer leaves the system. The queueing discipline in the background queue is also FCFS and the server does not pay any attention to the background queue as long as the foreground queue contains any customer. When a customer reaches the server for the second time he is once again given a quantum of service, the sizes of the quanta may vary, and if he needs more service he is once again sent back to the tail of the background queue. The generalization made is to allow, in theory, an infinite number of background queues and consequently let the quantum size shrink to zero. In order to be able to draw a figure of such a system, let us assume that we only allow a total of I queues in the system, i.e. one foreground queue, that can accept arrivals, and $I-1$ background queues, that will accept cycled customers, figure 3:6.

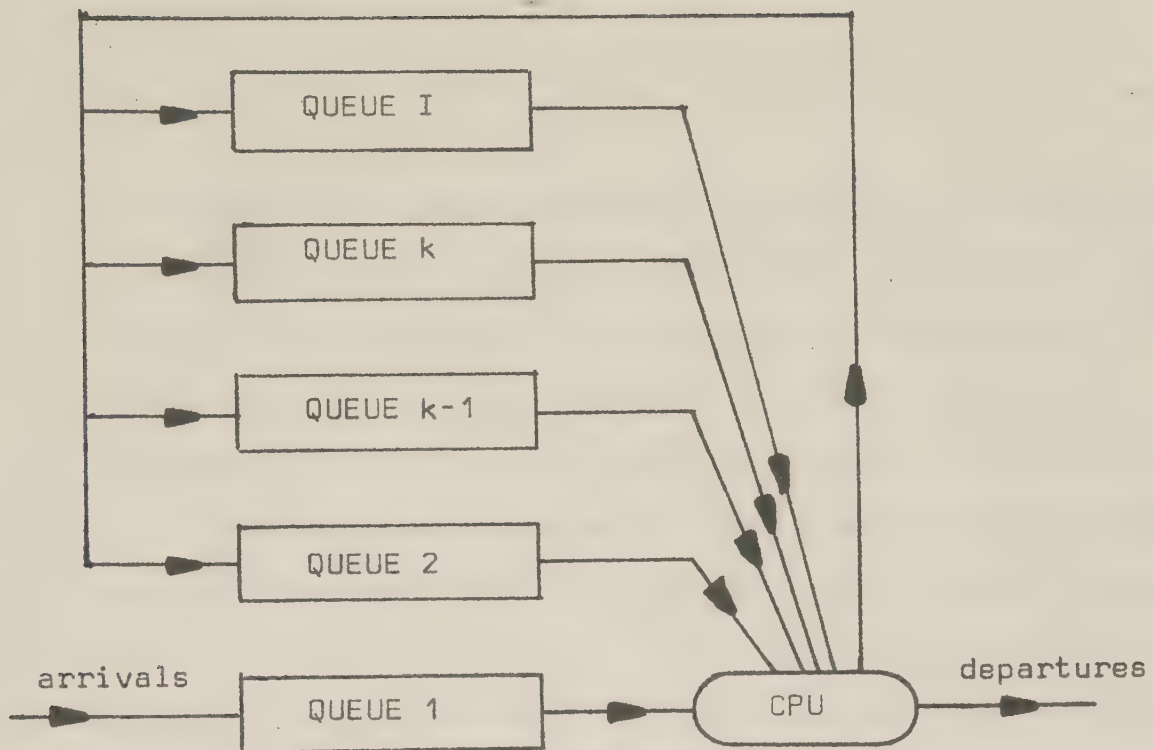


Fig 3:6

The system depicted in figure 3:6, might be called an FB-I system. The system operates as follows:

A customer that comes from the $k-1$: st queue is given a quantum of service, if he needs more service, he is sent to the tail of the k : th queue, where he has to wait for further service. When the server, actually the scheduler, selects a new customer for service the customer at the head of the lowest numbered non-empty queue is chosen. If we let I be infinitely large and at the same time make the quantum size infinitesimal we get a system that operates under the foreground background (FB) discipline. This implies among other things that a new customer is given immediate service, at least a small portion of service.

The conditional mean response time can now be calculated through the following arguments; Look at a customer that arrives to the system and whose service requirement is x seconds. This tagged job will possibly find other customers in the system. Those customers in the system with attained service greater than x seconds will obviously not interfere with the tagged job. Thus, from the tagged job's horizon any job that needs more than x seconds of service could as well leave the system after receiving x seconds of service. We therefore introduce the truncated service time distribution.

$$B_x(y) = \begin{cases} B(y) & y < x \\ 1 & y \geq x \end{cases} \quad (3.9)$$

The moments of this truncated distribution are defined as

$$\overline{x_x^n} = \int_0^x y^n dB(y) + x^n (1 - B(x)) \quad (3.10)$$

($n = 1, 2, \dots$)

Now consider an M/G/1 system, where service time is truncated at x , and the scheduling discipline FCFS. The average unfinished work in such a system would be

$$W_x = \frac{\frac{\lambda^2 \overline{x_x^2}}{2(1-\rho_x)}}{\quad} \quad (3.11)$$

where the utilization factor ρ_x is

$$\rho_x \triangleq \lambda \overline{x_x} \quad (3.12)$$

In our FB system the tagged job, on the average, has to be delayed by at least W_x . The conditional mean response time $T(x)$ is now calculated as

$T(x)$ = "x, the tagged job's service requirement" +
 "W_x, the mean unfinished work that the tagged job finds upon entry and which will interfere with him" + "delay caused by customers that arrive to the system while the tagged job is there"

We easily find that

$$T(x) = x + W_x + \lambda T(x) \overline{x_x} \quad (3.13)$$

since the customers that arrive to the system while the tagged job is there will delay him by $\overline{x_x}$ seconds on average.

Equation (3.13) yields

$$T(x) = \frac{W_x + x}{1 - \rho_x} \quad (3.14)$$

or

$$W(x) = \frac{W_x + x\rho_x}{1 - \rho_x} \quad (3.15)$$

3.5 Some Numerical Examples.

Let us by some examples illustrate the behaviour of $W(x)$ for the scheduling algorithms that so far have been treated. As the first example we choose an M/M/1 system, i.e.

$$B(x) = 1 - e^{-\mu x} \quad x \geq 0 \quad (3.16)$$

We then get with $\rho = \lambda/\mu$

$$W_{FCFS}(x) = \frac{\rho}{\mu(1-\rho)} \quad (3.17)$$

$$W_{RR}(x) = W_{LCFS-PR} = \frac{\rho x}{1-\rho} \quad (3.18)$$

$$W_{FB}(x) = \frac{\rho}{\mu} \frac{1 - e^{-\mu x} - \mu x e^{-\mu x}}{(1 - \rho(1 - e^{-\mu x}))^2} + \frac{\rho(1 - e^{-\mu x})x}{1 - \rho(1 - e^{-\mu x})} \quad (3.19)$$

Equations (3.17) - (3.19) are drawn in figure 3:7, where $\mu = 1$, and $\rho = 0.75$. We note the extreme fairness of the scheduling disciplines RR and LCFS-PR, the total ignorance of service requirement x by FCFS, and the tendency to favour short jobs that is inherent for FB - scheduling.

The second example is a system, where the service times are distributed according to an H_2 - distribution, i.e.

$$b(x) = \alpha \mu_1 e^{-\mu_1 x} + (1-\alpha) \mu_2 e^{-\mu_2 x} \quad x \geq 0 \quad (3.20)$$

for which

$$\bar{x} = \frac{\alpha}{\mu_1} + \frac{1-\alpha}{\mu_2} \quad (3.21)$$

and

$$\overline{x^2} = \frac{2\alpha}{\mu_1^2} + \frac{2(1-\alpha)}{\mu_2^2} \quad (3.22)$$

The explicit formulas for $W(x)$ are not reproduced here, but they can quite easily be computed. At least $W_{FCFS}(x)$,

$W_{RR}(x)$, and $W_{LCFS-PR}(x)$ are quite simple.

$W_{FB}(x)$, however, is a bit more involved but quite obtainable. Figure 3:8 shows a graph of the response curves when $\rho = 0.75$, $\bar{x} = 1$, $\alpha = 2/3$, and $\mu_2 = 2$.

Obviously the same comments apply to our second example as the comments about the first example.

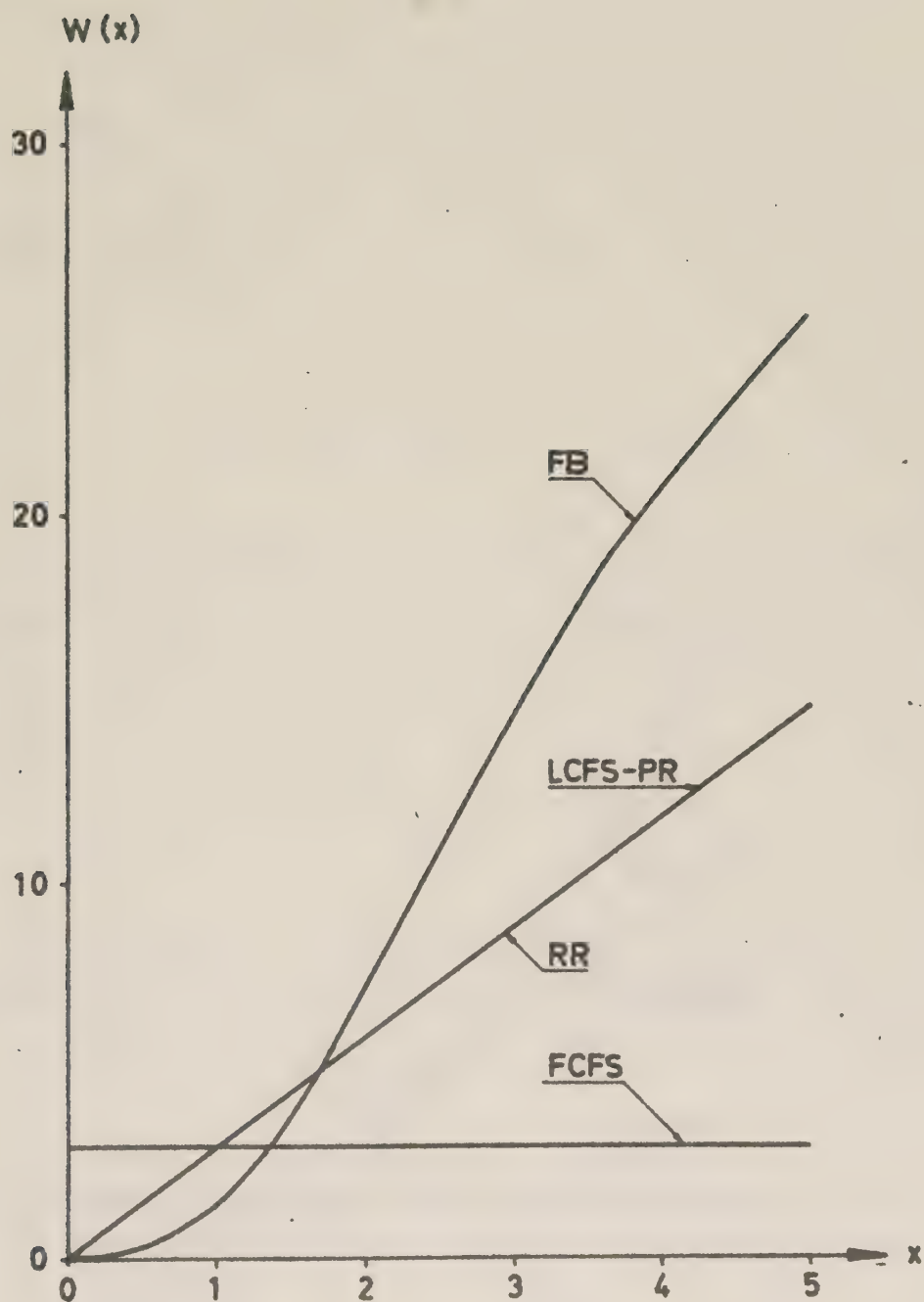


Fig. 3:7. Average conditional waiting time for the system $M/M/1$, $\rho = 0.75$ and $\mu = 1$.

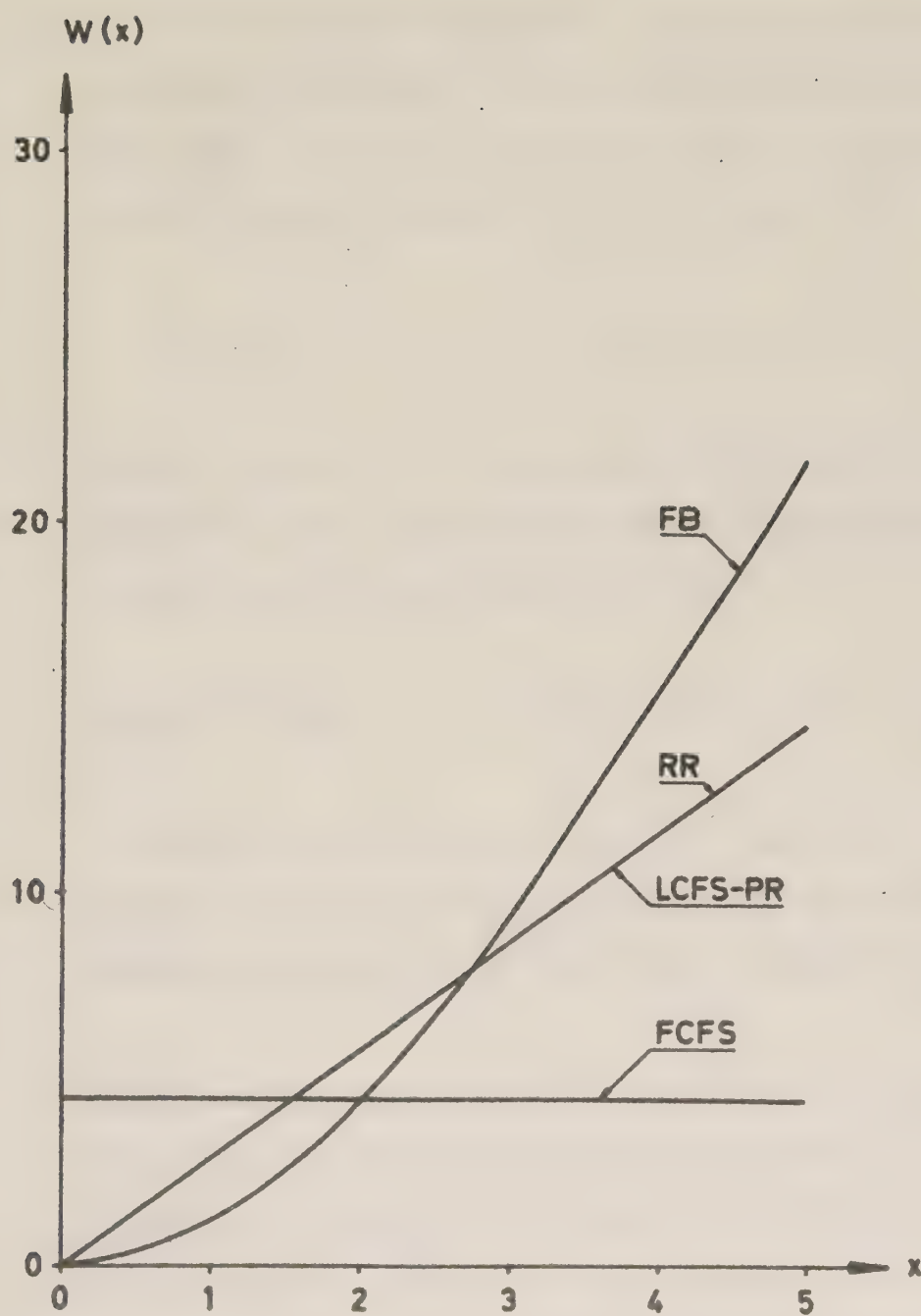


Fig. 3:8. Average conditional waiting time for an $M/H_2/1$ system, $\rho = 0.75$, $\mu_2 = 2$, $\nu_2 = 0.5$, and $\alpha = 2/3$.

4. The Multilevel (ML) Scheduling Algorithm.

In 1970 Kleinrock presented a scheduling algorithm which he called multilevel (ML) scheduling algorithm (KLEI 70b, KLEI 71b). The basic underlying idea is that we define a set of attained service times (a_i) such that

$$0 = a_0 < a_1 < a_2 < \dots < a_N < a_{N+1} = \infty \quad (4.1)$$

We also define $N + 1$ scheduling disciplines where the discipline for a job that so far has received τ seconds of service, in the interval (level)

$$a_{i-1} \leq \tau < a_i \quad i = 1, 2, \dots, N + 1 \quad (4.2)$$

is denoted by D_i . We have actually defined a system of queues, whose structure is similar to that of the FB-I system, see chapter 3. Kleinrock allowed D_i to be either:

FCFS (batch processing); FB; or RR. We are here going to permit yet another discipline namely LCFS - PR. In addition between the intervals the jobs are treated as a set of FB disciplines; that is, the processor will give its complete attention to those jobs in the lowest level non-empty queue and will schedule them according to the discipline appropriate for that interval. See figure 4:1.

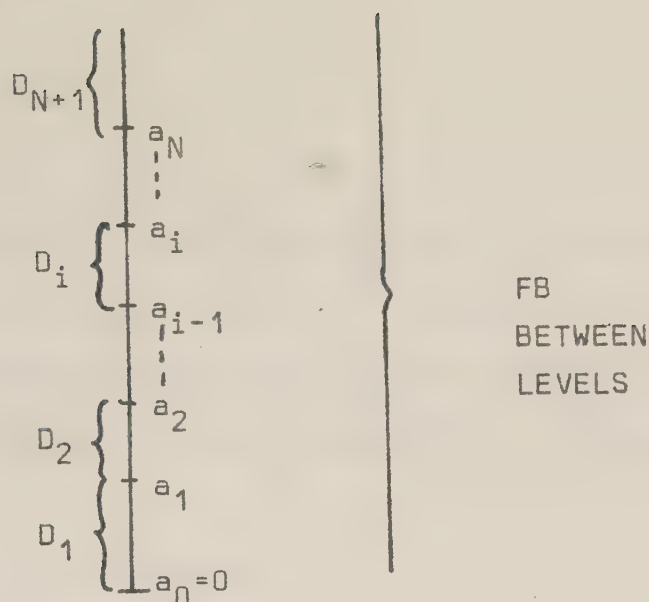


Fig. 4:1

If $N = 1$ we can have any of 16 disciplines (FCFS followed by FCFS, ---, LCFS - PR followed by LCFS - PR), note that FB followed by FB is just a single FB system.

The quantity we wish to solve for is $T(x)$ (sometimes $W(x)$).

We begin with the case where the i^{th} discipline is FCFS. Let us assume that

$$a_{i-1} \leq x < a_i \quad (4:3)$$

($D_i = \text{FCFS}$)

Kleinrock shows that

$$T(x) = \frac{W_{a_i} + x}{1 - \rho_{a_{i-1}}} \quad (4.4)$$

where as usual W_{a_i} is the mean unfinished work when the service time distribution is truncated at a_i . Now we assume that the i^{th} level is FB. Due to the overall FB policy we can look upon the system as if it were a pure FB system and consequently we find that

$$T(x) = \frac{W_x + x}{1 - \rho_x} \quad (4.5)$$

The case when the i^{th} level is RR is a bit more complex. If the customer departs during the first level; i.e.

$$0 \leq x < a_1 \quad (4.6)$$

then

$$T(x) = \frac{x}{1 - \rho_{a_1}} \quad (4.7)$$

which is just like a pure RR system where service time distribution is truncated at $x = a_1$.

When $i \geq 2$ we have to proceed more carefully and Kleinrock shows that, (KLEI 75b)

if

$$x = a_{i-1} + \tau \quad (4.8)$$

we can compute $T(x)$ as

$$T(x) = T(a_{i-1} + \tau) = \frac{1}{1 - \rho_{a_{i-1}}} \{W_{a_{i-1}} + a_{i-1} + \alpha_2(\tau)\} \quad (4.9)$$

Where $\alpha_2(\tau)$ is an unknown function that has to be determined. To solve for $\alpha_2(\tau)$ we must, in general, consider an M/G/1 system with bulk arrivals and RR discipline.

In (KLEI 71b, 75b) it is shown that $\alpha_2(\tau)$ is determined by the following integral equation. We write $\alpha(\tau)$ instead of $\alpha_2(\tau)$

$$\begin{aligned} \frac{d\alpha}{d\tau} = & \lambda \bar{a} \int_0^{\infty} \frac{d\alpha}{dx} (1 - F(\tau + x)) dx + \\ & + \lambda \bar{a} \int_0^{\infty} \frac{d\alpha}{dx} (1 - F(\tau - x)) dx + 1 + b(1 - F(\tau)). \end{aligned} \quad (4.10)$$

where

$$F(x) = \begin{cases} \frac{B(a_{i-1} + x) - B(a_{i-1})}{1 - B(a_{i-1})} & 0 \leq x < a_i - a_{i-1} \\ 1 & x > a_i - a_{i-1} \end{cases} \quad (4.11)$$

and

$\bar{a}$ = mean group size

b = mean number of customers that arrive together with the tagged job to the i^{th} level

Defining the generating function $H(z)$ for the bulk size $\tilde{h}$ defined as

$$H(z) = \sum_{k=0}^{\infty} P(\tilde{h}=k) z^k \quad (4.12)$$

we know at once that

$$\bar{a} = \left. \frac{dH}{dz} \right|_{z=1} \quad (4.13)$$

The value of b can also be expressed in terms of $H(z)$ and we find

$$b = \left. \frac{d^2 H}{dz^2} \right|_{z=1} \frac{1}{\bar{a}} \quad (4.14)$$

What remains to be done is to solve the integral equation. In (KLEI 71b) the solution was found when

$$1-F(\tau) = \begin{cases} q(\tau) e^{-\mu\tau} & 0 \leq \tau < a_i - a_{i-1} = x_1 \\ 0 & x_1 \leq \tau \end{cases} \quad (4.15)$$

where $q(\tau)$ is a polynomial of degree n (restricted only in that it must render $B(\tau)$ a distribution function).

This $1-F(\tau)$ covers the case when the service time distribution is Erlang- n , that is

$$b(x) = n\mu \frac{(n\mu x)^{n-1}}{(n-1)!} e^{-n\mu x} \quad (4.16)$$

The solution to equation (4.10) is rather involved and it can be written as

$$\frac{d\alpha}{d\tau} = \frac{1}{1-\lambda \bar{a} \bar{\tau}} - \frac{b}{\lambda \bar{a}} \sum_{i=1}^{n+1} \frac{(\mu^2 - \gamma_i^2)^{n+1}}{Q_2^{(1)}(\gamma_i)} \frac{Q_0(\gamma_i) e^{-\gamma_i \tau} - Q_1(\gamma_i) e^{-\gamma_i (x_1 - \tau)}}{Q_0(\gamma_i) + Q_1(\gamma_i) e^{-\gamma_i x_1}} \quad (4.17)$$

where

$$Q_0(y) = (y+\mu)^{n+1-\lambda \bar{a}} \sum_{i=0}^n q^{(i)}(0) (y+\mu)^{n-i} \quad (4.18)$$

$$Q_1(y) = \lambda \bar{a} \sum_{i=0}^n e^{-\mu x_1} q^{(i)}(x_1) (y+\mu)^{n-i} \quad (4.19)$$

$$Q_2(y) = Q_0(y) Q_0(-y) - Q_1(y) Q_1(-y) \quad (4.20)$$

and where the roots of the equation

$$Q_2(y) = 0 \quad (4.21)$$

occur in pairs which we denote by $(-\gamma_i, +\gamma_i)$ for $i=1, 2, \dots, n+1$. $\bar{\tau}$ is the expectation of $F(\tau)$, i.e.

$$\bar{\tau} = \int_0^{x_1} (1-F(\tau)) d\tau \quad (4.22)$$

The expression for $1-F(\tau)$ as given by equation (4.15) does not include the case when the service times are distributed as an H_2 - distribution, i.e.

$$b(x) = \alpha \mu_1 e^{-\mu_1 x} + (1-\alpha) \mu_2 e^{-\mu_2 x} \quad (4.23)$$

If we want to derive $a(\tau)$ for such a service time distribution, we have to solve (4.10) when $1-F(\tau)$ is of the form

$$1-F(\tau) = \begin{cases} \epsilon e^{-\mu_1 \tau} + (1-\epsilon) e^{-\mu_2 \tau} & 0 \leq \tau < a_1 - a_{1-1} = x_1 \\ 0 & \tau \geq x_1 \end{cases} \quad (4.24)$$

where ϵ can be determined from equations (4.11) and (4.23)

If we use $y(\tau) = \frac{da}{d\tau}$, equation (4.10) can be written as

$$\begin{aligned} y(\tau) = & \lambda \bar{a} \epsilon e^{-\mu_1 \tau} \left\{ \int_0^{x_1} y(\tau) e^{-\mu_1 \tau} d\tau - \int_0^{\tau} g(u, \mu_1) du \right\} + \\ & + \lambda \bar{a} (1-\epsilon) e^{-\mu_2 \tau} \left\{ \int_0^{x_1} y(\tau) e^{-\mu_2 \tau} d\tau - \int_0^{\tau} g(u, \mu_2) du \right\} + \\ & + 1 + b(\epsilon e^{-\mu_1 \tau} + (1-\epsilon) e^{-\mu_2 \tau}) \end{aligned} \quad (4.25)$$

where

$$g(u, \mu) = (y(x-u)e^{-\mu x_1} - y(u))e^{\mu u} \quad (4.26)$$

Differentiating (4.25) twice with respect to τ and forming

$$\frac{d^2 y}{d\tau^2} + (\mu_1 + \mu_2) \frac{dy}{d\tau} + \mu_1 \mu_2 y = \text{----} \quad (4.27)$$

will lead to, with $D = \frac{d}{d\tau}$

$$\begin{aligned} & (D^2 + (\mu_1 + \mu_2)D + \mu_1 \mu_2 - \lambda \bar{a}(D + \mu_1(1-\epsilon) + \mu_2 \epsilon))y(\tau) + \\ & + \lambda \bar{a}((\epsilon e^{-\mu_1 x_1} + (1-\epsilon)e^{-\mu_2 x_1})D + \\ & + \epsilon \mu_2 e^{-\mu_1 x_1} + (1-\epsilon)\mu_1 e^{-\mu_2 x_1})y(x_1 - \tau) = \mu_1 \mu_2 \end{aligned} \quad (4.28)$$

or

$$Q_0(D)y(\tau) + Q_1(D)y(x_1 - \tau) = \mu_1 \mu_2 \quad (4.29)$$

where

$$Q_0(D) = (D + \mu_1)(D + \mu_2) - \lambda \bar{a}(D + \mu_1(1-\epsilon) + \mu_2 \epsilon) \quad (4.30)$$

$$\begin{aligned} Q_1(D) = & \lambda \bar{a}((\epsilon e^{-\mu_1 x_1} + (1-\epsilon)e^{-\mu_2 x_1})D + \\ & + \epsilon \mu_2 e^{-\mu_1 x_1} + (1-\epsilon)\mu_1 e^{-\mu_2 x_1}) \end{aligned} \quad (4.31)$$

Replacing τ by $x_1 - \tau$ in equation (4.29) we get

$$Q_0(-D) y(x_1 - \tau) + Q_1(-D) y(\tau) = \mu_1 \mu_2 \quad (4.32)$$

which after combining (4.29) and (4.32) leads to

$$Q_2(D) y(\tau) = \{Q_0(0) - Q_1(0)\} \mu_1 \mu_2 \quad (4.33)$$

where

$$Q_2(D) = Q_0(D) Q_0(-D) - Q_1(D) Q_1(-D) \quad (4.34)$$

$Q_2(D)$ is a polynomial of degree 4. Its roots occur in pairs $(-\gamma_1, \gamma_1)$ $(-\gamma_2, \gamma_2)$ and they are distinct and non-zero.

The general solution to (4.33) is

$$y(\tau) = c_0 + \sum_1^2 c_k (Q_0(\gamma_k) e^{-\gamma_k \tau} - Q_1(\gamma_k) e^{-\gamma_k (x_1 - \tau)}) \quad (4.35)$$

where the unknown coefficients c_0, c_1, c_2 are to be determined. c_0 is readily obtained as

$$c_0 = \frac{Q_0(0) - Q_1(0)}{Q_2(0)} \mu_1 \mu_2 \quad (4.36)$$

which reduces to

$$c_0 = \frac{1}{1 - \lambda \bar{a} \bar{\tau}} \quad (4.37)$$

where

$$\bar{\tau} = \frac{\epsilon}{\mu_1} (1 - e^{-\mu_1 x_1}) + \frac{1 - \epsilon}{\mu_2} (1 - e^{-\mu_2 x_1}) \quad (4.38)$$

The expression for $y(\tau)$ must be put in the original integral equation to determine c_1 and c_2 . Collecting the coefficients of the various exponentials that arise, we get

$$c_k = - \frac{b}{\lambda \bar{a}} \frac{(\gamma_k^2 - \mu_1^2)(\gamma_k^2 - \mu_2^2)}{Q_2^1(\gamma_k)(Q_0(\gamma_k) + Q_1(\gamma_k)e^{-\gamma_k x_1})} \quad (k=1,2) \quad (4.39)$$

The solution can then be written as

$$\frac{d\alpha}{d\tau} = \frac{1}{1 - \lambda \bar{a} \bar{\tau}} - \frac{b}{\lambda \bar{a}} \frac{\sum_k \frac{2(\gamma_k^2 - \mu_1^2)(\gamma_k^2 - \mu_2^2)(Q_0(\gamma_k)e^{-\gamma_k \tau} - Q_1(\gamma_k)e^{-\gamma_k(x_1 - \tau)})}{Q_2^1(\gamma_k)(Q_0(\gamma_k) + Q_1(\gamma_k)e^{-\gamma_k x_1})}}{1} \quad (4.40)$$

It is quite interesting to compare the solution obtained by Kleinrock et al (4.17) and the solution we found (4.40).

The main structure of the solutions is the same and we can obtain Kleinrock's solution, when $q(\tau) = 1$, if we put $\mu_1 = \mu_2 = \mu$ in equation (4.40). This might perhaps lead us to suspect that it is possible to solve the integral equation (4.10), when $1-F(\tau)$ has a more complicated structure. Below we shall prove that this indeed is possible and we will show how to solve the integral equation- albeit the solution is very complex - when $1-F(\tau)$ is of the form

$$1-F(\tau) = \sum_{v=1}^R q_v(\tau) e^{-\mu_v \tau} \quad 0 \leq \tau < x_1 \quad (4.41)$$

where $q_v(\tau)$ is a polynomial in τ of degree k_v , the only restriction placed on the $q_v(\tau)$'s is that they should render $B(\tau)$ a distribution function. Obviously we must still claim that

$$1-F(\tau) = 0 \text{ when } \tau \geq x_1 \quad (4.42)$$

Furthermore $R \geq 1$ and $\mu_v \geq 0 \quad v = 1, 2, \dots, R$.

We note that if we can solve the integral equation for this type of $1-F(\tau)$ we have in effect found the solution for all the cases for which $B^*(s)$ is a rational function of s .

Using $y(\tau)$ instead of $\frac{d\alpha}{d\tau}$ we can rewrite the integral equation as

$$y(\tau) = \lambda \bar{a} \sum_{v=1}^R e^{-\mu_v \tau} g(\tau, \mu_v) + 1 +$$

$$+ b \sum_{v=1}^R q_v(\tau) e^{-\mu_v \tau} \quad (4.43)$$

where

$$g(\tau, \mu_v) = \int_0^{x_1 - \tau} y(u) q_v(\tau + u) e^{-\mu_v u} du +$$

$$+ \int_0^{\tau} y(u) q_v(\tau - u) e^{\mu_v u} du \quad (4.44)$$

We now introduce the differential polynomial $\psi(D)$, where $D = \frac{d}{d\tau}$

$$\psi(D) = \prod_{v=1}^R (D + \mu_v)^{k_v + 1} \quad (4.45)$$

Let $\psi(D)$ operate on equation (4.43) we then find that

$$\psi(D) y(\tau) = \sum_{v=1}^R \phi_v(D) \left\{ \sum_{i=0}^{k_v} q_v^{(i)}(0) (D + \mu_v)^{k_v - i} y(\tau) - \right.$$

$$\left. - e^{-\mu_v x_1} \sum_{i=0}^{k_v} q_v^{(i)}(x_1) (D + \mu_v)^{k_v - i} y(x_1 - \tau) \right\} + \psi(0)$$

$$(4.46)$$

where

$$\phi_u(D) = \prod_{\substack{r=1 \\ r \neq u}}^R (D + \mu_r)^{k_r+1} \quad (4.47)$$

and

$$\psi(D) = \prod_{u=1}^R \mu_u^{k_u+1} \quad (4.48)$$

Next we introduce the polynomials $Q_0(D)$ and $Q_1(D)$ which enable us to write equation (4.46) as

$$Q_0(D) y(\tau) + Q_1(D) y(x_1 - \tau) = \psi(D) \quad (4.49)$$

where $Q_0(D)$ and $Q_1(D)$ can be found from equation (4.46).

The procedure is now the same as in the previous solutions, eliminate $y(x_1 - \tau)$ by replacing τ with $x_1 - \tau$ and manipulate the equation somewhat. Finally we arrive at

$$Q_2(D) y(\tau) = \{Q_0(D) - Q_1(D)\} \psi(D) \quad (4.50)$$

where as usual

$$Q_2(D) = Q_0(D) Q_0(-D) - Q_1(D) Q_1(-D) \quad (4.51)$$

Equation (4.51) is a linear differential equation of degree

$$2 \sum_{v=1}^R (k_v + 1) \quad (4.52)$$

The solution to this equation is straightforward, if the roots to the equation

$$Q_2(y) = 0 \quad (4.53)$$

are simple, -we know that they must occur in pairs. We feel, however, that no further insight may be gained by going through all the complicated calculations and therefore we choose to end the presentation at this point.

If the discipline employed in the i^{th} level is LCFS-PR, the analysis will be somewhat easier than that for the RR discipline. Actually, this analysis can be performed quite easily, using simple arguments. Let us first use a more detailed method, and - when that has been done - let's try the easy way of performing the analysis.

If the i^{th} level is also the first level, then the mean conditional response time for a customer requiring x seconds of service, where $0 \leq x < a_i$, will be

$$T(x) = \frac{x}{1-\rho a_1} \quad (4.54)$$

This is obviously the same result as for a pure LCFS-PR system, where the service time distribution is truncated at $x = a_1$.

When the i^{th} level uses a LCFS-PR discipline and $i \geq 2$ the analysis becomes a bit harder. One way of performing the analysis is to look upon the system as if it were a multi-level system with two breakpoints a_{i-1} and a_i .

Figure 4:2

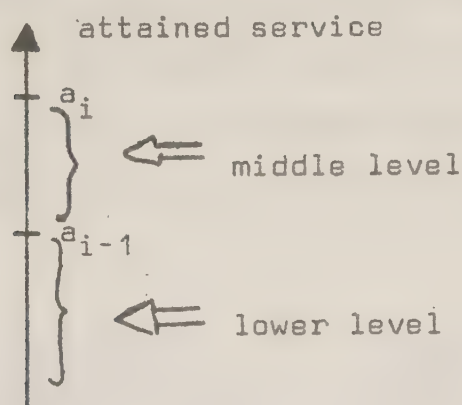


Fig.4:2

Due to the overall FB policy the processing of work takes place in the lower level, as soon as there is any customer with attained service less than a_{i-1} . As soon as the lower level is emptied of customers processing may take place in the middle level. Obviously there may be a number of customers waiting at the threshold to the middle level. Now we assume that the tagged job has a service requirement x , that can be written as

$$x = a_{i-1} + \tau \quad (4.55)$$

where $\tau < a_i - a_{i-1}$

The time intervals when work is processed at the lower level are called lower level busy periods (l.l.b.p.) see figure 4:3. During each such l.l.b.p. a number of customers may reach the threshold to the middle level. We choose to consider those customers as arriving at the end of the lower level busy period. This assumption does not change the structure of the queueing discipline since those customers that arrive at the threshold during the l.l.b.p. have to wait for the end of the l.l.b.p. before they can get service in the middle level. A time diagram for the service system may be depicted as in figure 4:3

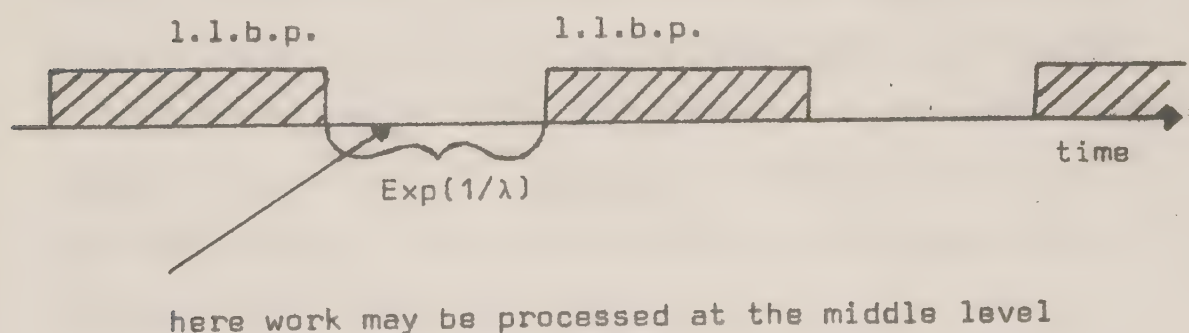


Fig. 4:3

The idle period distribution for the lower level is an exponential distribution with parameter λ , since as soon as there is an arrival to the system a l.l.b.p. is started. We now create a virtual time axis in such a way that the l.l.b.p.'s are removed from the time axis, figure 4:4

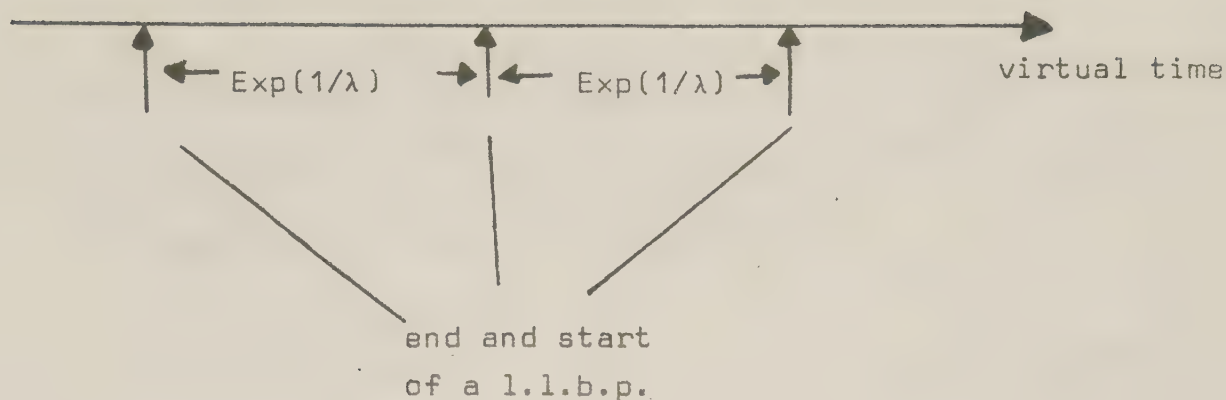


Fig 4:4.

At those points of virtual time, where the l.l.b.p.:s end and start, we have a bulk arrival to the middle level, We are therefore forced to study an M/G/1 system with bulk arrivals.

Now let us determine the mean conditional response time for the tagged job. Let α_1 be the mean real time the job spends in the system until its arrival at the middle level queue. Let $\alpha_2(\tau)$ be the virtual time the job spends in the middle level. The initial delay the job experiences is equal to the mean work the job finds in the lower level on arrival plus the a_{i-1} seconds of work that it contributes to the lower level. During its stay in the lower level other jobs may arrive to the system and thus contribute to the time the tagged job spends in the lower level. The mean number of the other jobs that arrive to

the system during this time is $\lambda \alpha_1$ and they will, on average, produce a delay of $\lambda \alpha_1 \bar{x}_{a_{i-1}}$, therefore

$$\alpha_1 = W_{a_{i-1}} + a_{i-1} + \lambda \alpha_1 \bar{x}_{a_{i-1}} \quad \text{which gives}$$

$$\alpha_1 = \frac{W_{a_{i-1}} + a_{i-1}}{1 - \rho_{a_{i-1}}} \quad (4.56)$$

$\alpha_2(\tau)$ the mean virtual time the job spends in the middle level, can easily be converted into real time spent in this level. In the virtual time $\alpha_2(\tau)$ there are an average of $\lambda \alpha_2(\tau)$ l.l.b.p.'s that have been ignored. Each of these have a mean length of $\bar{x}_{a_{i-1}} / (1 - \rho_{a_{i-1}})$ (KLEI 75b) Therefore the mean real time the job spends in the middle level is given by

$$\alpha_2(\tau) + \lambda \alpha_2(\tau) \frac{\bar{x}_{a_{i-1}}}{1 - \rho_{a_{i-1}}} = \frac{\alpha_2(\tau)}{1 - \rho_{a_{i-1}}} \quad (4.57)$$

Combining equations (4.56) and (4.57) we see that a job requiring $x = a_{i-1} + \tau$ seconds of service has a mean response time given by

$$T(a_{i-1} + \tau) = \frac{1}{1 - \rho_{a_{i-1}}} \{W_{a_{i-1}} + a_{i-1} + \alpha_2(\tau)\} \quad (4.58)$$

Equation (4.58) has the same structure as equation (4.9), that was derived for RR scheduling. We even have the same problem left, namely to determine $\alpha_2(\tau)$. For RR scheduling we were able to determine $\alpha_2(\tau)$ by studying an M/G/1 system with bulk arrivals and RR discipline. Here we study an M/G/1 system with bulk arrivals and LCFS-PR discipline. When the tagged job arrives at the middle level, this event is assumed to take place at the end of a l.l.b.p., he is accompanied by some other jobs. Due to the LCFS-PR discipline only some of those accompanying jobs will interfere with the tagged job in the middle level. The average number of jobs that will delay our job is denoted by b , and can be written as

$$b = \lambda \alpha_1 (1 - B(a_{i-1})) \quad (4.59)$$

$\lambda \alpha_1$ is the mean number of jobs that arrive to the system, while the tagged job is in the lower level. $1 - B(a_{i-1})$ is the fraction of those jobs that need more than a_{i-1} seconds of service and therefore will accompany the tagged job into the middle level.

Now we analyse an M/G/1 queueing system with group arrivals and queueing discipline LCFS-PR: We are especially interested in the busy period distribution for such a system. Let us assume that the group size distribution

is defined through

$$P(\text{group size} = k) = h_k \quad (4.60)$$

and let

$$H(z) = \sum_{k=0}^{\infty} h_k z^k \quad (4.61)$$

Suppose that a busy period is started by a single customer.

Let

Y Δ length of a busy period, that is initiated by a single customer.

$x_1 \Delta$ service requirement for that customer. (4.62)

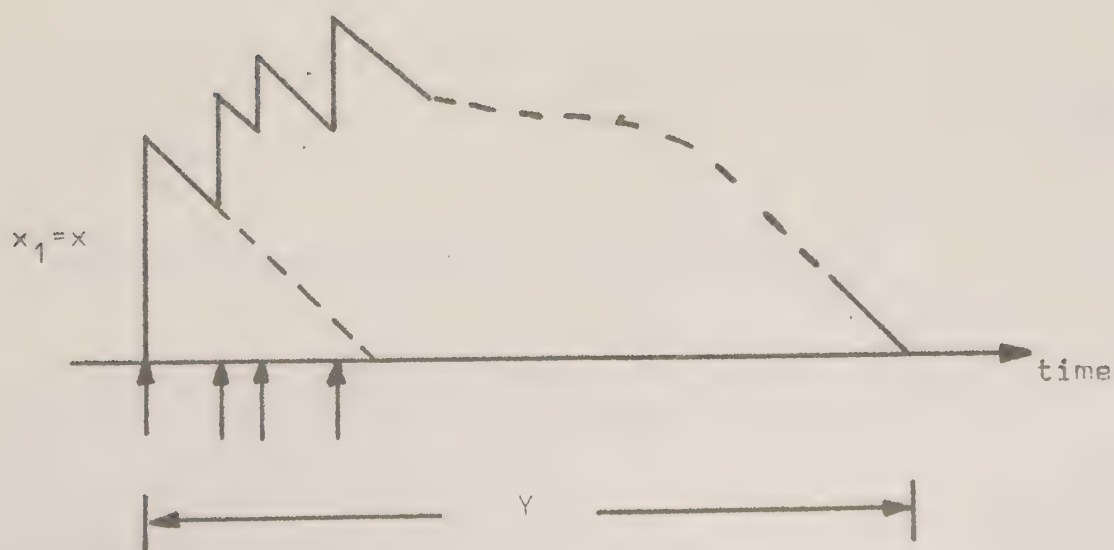


Fig. 4:5

During the service of the first customer, the one that initiated the busy period, a number of arrivals may occur. Each one of those arrivals consists of a group of customers. Clearly due to the LCFS-PR discipline the first customer will be preempted if an arrival occurs. We are, however, only interested in the busy period distribution. Thus the choice of queueing discipline is irrelevant and we may use any work conserving discipline, that operates independent of service time. Consequently we can invoke the classical busy period analysis, that involves sub busy periods etc.

Let $X_{i,j}$ = length of sub b.p., that is generated by the j^{th} customer in the i^{th} group (4.63)

Clearly Y and $X_{i,j}$ must be equal in distribution.

Now we form the following conditional expectation

$$E[e^{-sY} | x_1=x, \text{ n arrivals, group size} = k_i.] \quad (4.64)$$

$$i = 1, 2, \dots, n$$

where Y can be written as

$$Y = x + X_{1,1} + X_{1,2} + \dots + X_{1,k_1} + X_{2,1} + \dots + X_{n,k_n} \quad (4.65)$$

If we introduce

$$G^*(s) = E\{e^{-sY}\} = E\{e^{-sX_{i,j}}\} \quad (4.66)$$

the expectation (4.64) can be seen to equal

$$e^{-sx} (G^*(s))^{k_1+k_2+\dots+k_n} \quad (4.67)$$

Removing the condition on group size, we get

$$E\{e^{-sY} | X_1 = x, n \text{ arrivals}\} = e^{-sx} \{H(G^*(s))\}^n \quad (4.68)$$

Finally we get after removing the conditions on x_1 and the number of arrivals

$$G^*(s) = B^*(s + \lambda - \lambda H(G^*(s))) \quad (4.69)$$

Equation (4.69) is the Laplace transform of the density function for the length of a busy period, that is initiated by a single customer. The mean length of the busy period is easily calculated and it reads

$$\bar{g} = \text{"mean length"} = \frac{\bar{x}}{1 - \lambda \bar{x} \bar{a}} \quad (4.70)$$

where $\bar{a}$ is the mean group size and as usual $\bar{x}$ is the

mean service time.

If we knew that the service requirement for the customer, that initiates the busy period is exactly x seconds we can easily show that the mean length of the busy period is

$$\frac{x}{1 - \lambda \bar{x} \bar{a}} \quad (4:71)$$

Furthermore if the b.p. were initiated by a group of customers of size n , then the mean length of the resulting busy period would be

$$n \cdot \frac{\bar{x}}{1 - \lambda \bar{x} \bar{a}} \quad (4:72)$$

Returning now to the original problem of computing $\alpha_2(\tau)$, let us remember that we are working in virtual time! $\alpha_2(\tau)$ can then be expressed as

$\alpha_2(\tau)$ = "mean time to empty the middle level of the accompanying customers" + "mean time spent by the tagged job in order to receive τ seconds of service when it has been admitted to the server" (4:73)

Now using the equations (4.71) and (4.72) we easily get

$$\alpha_2(\tau) = \frac{b \bar{x}_0}{1 - \lambda \bar{x}_0 \bar{a}} + \frac{1}{1 - \lambda \bar{x}_0 \bar{a}} \quad (4.74)$$

where $\bar{x}_0$ can be computed as

$$\bar{x}_0 = \frac{1}{1 - B(a_{i-1})} \int_0^{a_i - a_{i-1}} (1 - B(a_{i-1} + x)) dx \quad (4.75)$$

The mean group size $\bar{a}$ can be calculated and the result is:

$$\bar{a} = \frac{1 - B(a_{i-1})}{1 - \rho_{a_{i-1}}} \quad (4.76)$$

since the mean number of served customers during a lower level busy period is

$$\frac{1}{1 - \rho_{a_{i-1}}} \quad (4.77)$$

and only the fraction $1 - B(a_{i-1})$ of those customers will need more than a_{i-1} seconds of service.

In summary the mean conditional response time can be written as

$$T(a_{i-1} + \tau) = \frac{1}{1 - \rho_{a_{i-1}}} \left\{ W_{a_{i-1}} + a_{i-1} + \frac{b \bar{x}_0 + \tau}{1 - \lambda \bar{x}_0 \bar{a}} \right\} \quad (4.78)$$

where $\bar{x}_0$ is determined through (4.75), and $\bar{a}$ is given by (4.76). b can be written as

$$b = \lambda \frac{W_{a_{i-1}} + a_{i-1}}{1 - \rho_{a_{i-1}}} (1 - B(a_{i-1})) \quad (4.79)$$

It turns out that equation (4.78) reduces to

$$T(x) = \frac{1}{1 - \rho_{a_i}} \{ W_{a_{i-1}} + x \} \quad (4.80)$$

This is obviously a much nicer expression than (4.78).

Below we will show how to derive (4.80) in an easier manner. This derivation is as follows:

$W(x)$ must be equal to an initial delay ($W_{a_{i-1}}$) plus a term equal to the mean delay due to customers arriving to the system while the tagged job is in the system.

This mean delay can be written as $\lambda T(x) \bar{x}_{a_i}$.

Remember that $D_i = \text{LCFS-PR}$. Thus

$$T(x) = W_{a_{i-1}} + x + \lambda T(x) \bar{x}_{a_i} \quad (4.81)$$

which reduces to equation (4.80)

Now let us demonstrate through some examples, the behaviour of the multilevel scheduling algorithms. We begin with three examples from the system $M/M/1$. Our first example is the iterated structure $D_1 = FCFS$. We choose $\mu = 1$, $\lambda = 0.75$, and $a_i = i$. Also in figure 4:6 we have superimposed a dashed line corresponding to the FB system over the entire range. In figure 4:7 we depict $W(x)$ for a multilevel structure with $N=3$, $a_i=i$ ($i=1,2,3$), and $D_1 = FCFS$, $D_2 = RR$, $D_3 = FB$, and $D_4 = FCFS$. Once again the parameters chosen in the $M/M/1$ system are $\mu = 1$, and $\lambda = 0.75$. The third and last example for the $M/M/1$ system has a similar structure to the one chosen for example 2. The only change made is that $D_2 = LCFS - PR$ otherwise the parameters are the same, see Figure 4:8

We leave $M/M/1$ now and give two examples for $M/G/1$. The first example is the system $M/E_2/1$ with $N = 2$. We choose to depict two queuing structures in Figure 4:9. The solid line corresponds to the case for which $D_1=D_2 = RR$, and $D_3 = FCFS$. The dashed line is the case where $D_1=D_2=LCFS-PR$, and $D_3 = FCFS$. $a_i = i$ ($i = 1,2$) and the parameters chosen are, once again $\mu = 1$, and $\lambda = 0.75$. Figure 4:9 gives us an opportunity to make a comparison between RR and $LCFS-PR$. The last example is for an $M/H_2/1$ system where we use the same queuing structures as in the previous example and also choose the parameters such that $\bar{x} = 1$, and $\lambda = 0.75$. See figure 4:10.

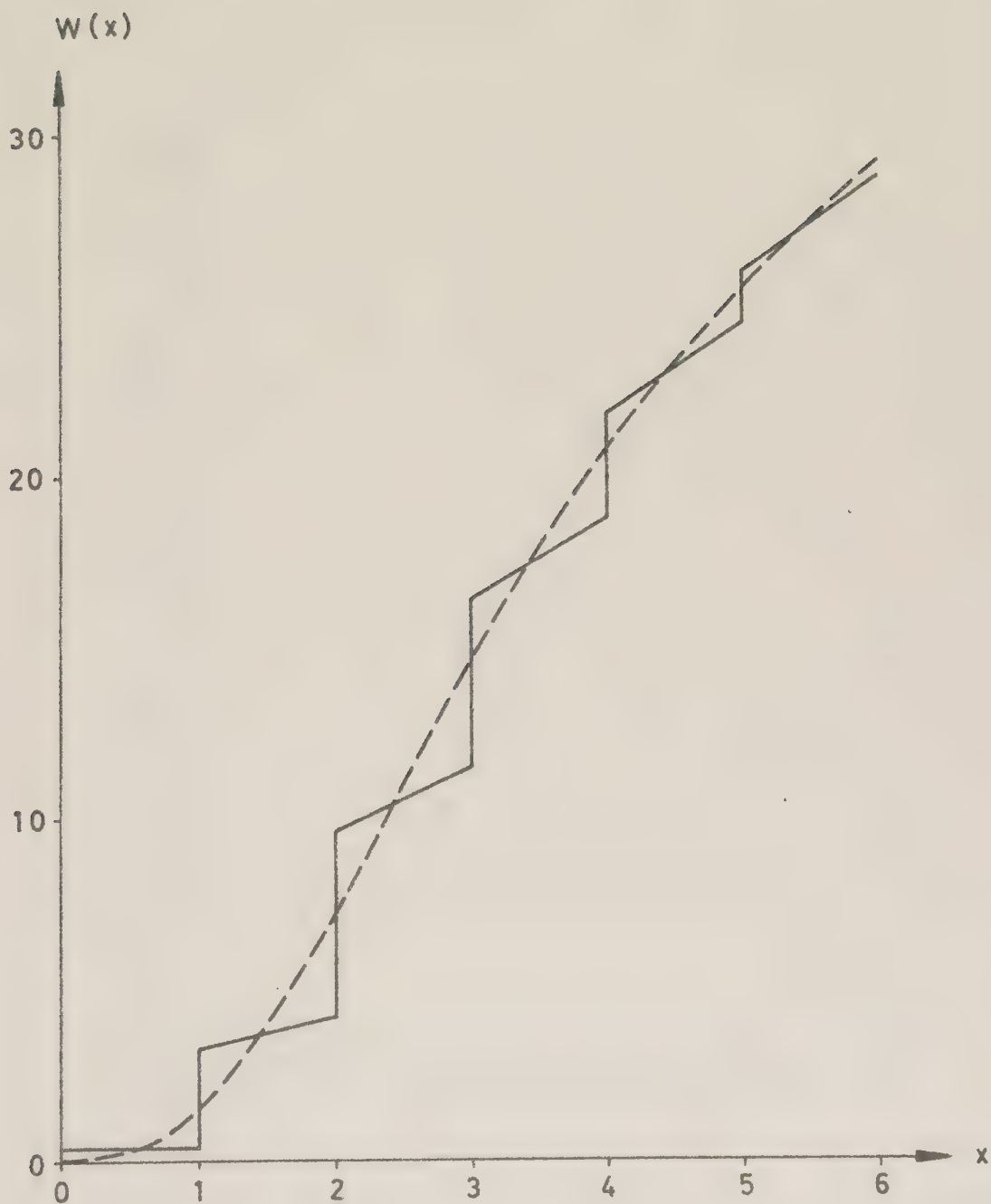


Fig. 4:6. Response time for an example of $N = \infty$,
 $D_i = \text{FCFS}$, $M/M/1$, $\bar{x} = 1$, $\lambda = 0.75$, and $a_i = i$.

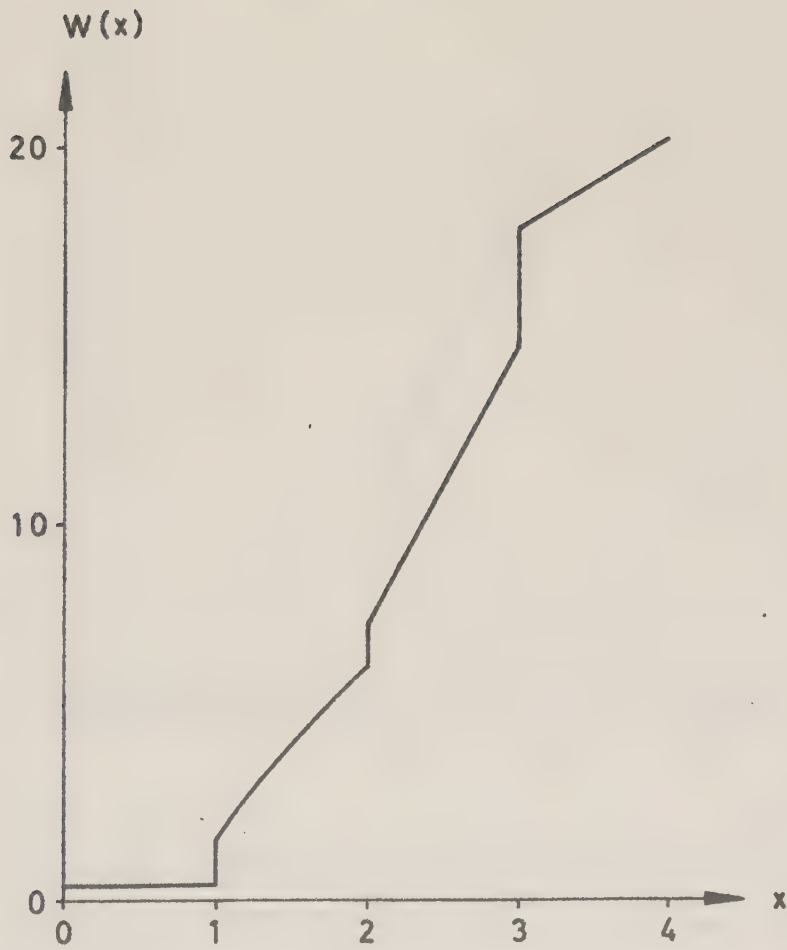


Fig. 4:7. Response time for $N = 3$, $a_i = i$, $D_1 = \text{FCFS}$, $D_2 = \text{RR}$, $D_3 = \text{FB}$, and $D_4 = \text{FCFS}$, $M/M/1$, $\bar{x} = 1$, and $\lambda = 0.75$.

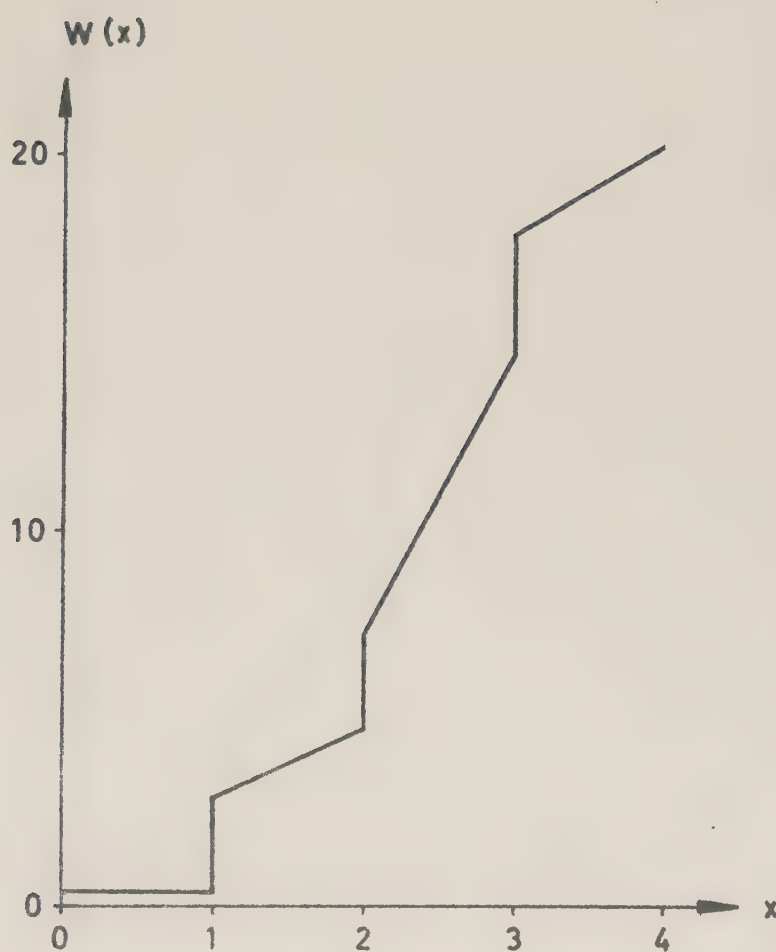


Fig. 4:8. Response time for an example of $N = 3$, $a_i = i$, $D_1 = \text{FCFS}$, $D_2 = \text{LCFS-PR}$, $D_3 = \text{FB}$, and $D_4 = \text{FCFS}$, $M/M/1$, $\bar{x} = 1$, and $\lambda = 0.75$.

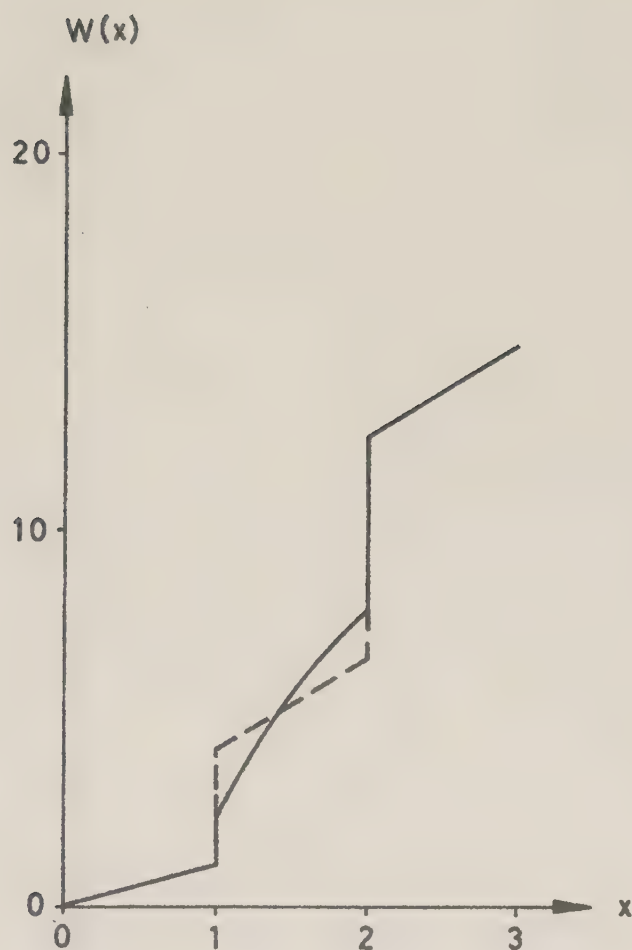


Fig. 4:9. Response times for $N = 2$ $a_i = i$, $M/E_2/1$,
 $\bar{x} = 1$, $\lambda = 0.75$, (— $D_1=D_2=RR$, and $D_3=FCFS$),
 (— — $D_1=D_2=LCFS-PR$; and $D_3=FCFS$)

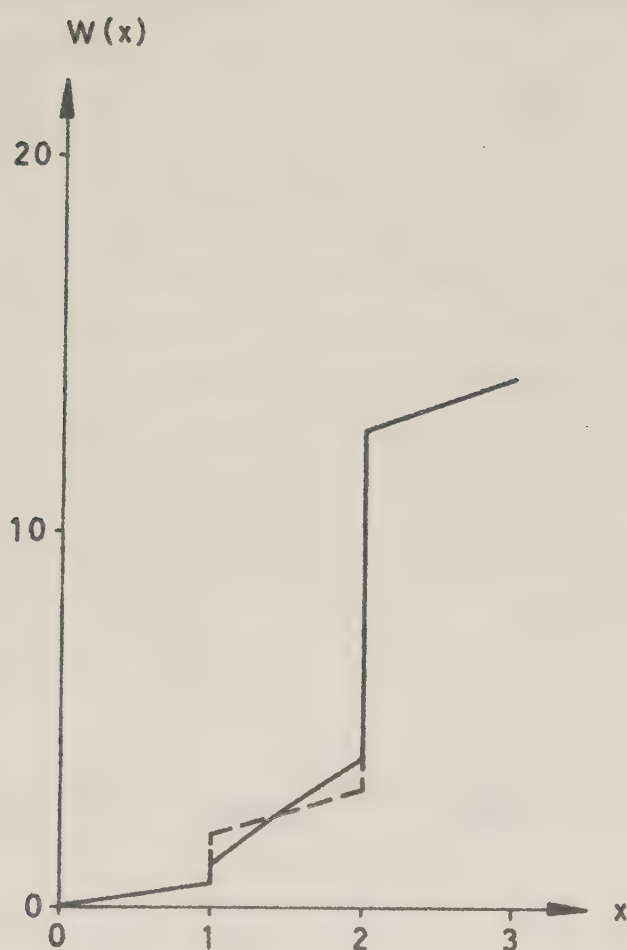


Fig. 4:10. Response times for $N=2$ $a_i=i$, $M/H_2/1$, $\bar{x}=1$, $\lambda=0.75$, (— $D_1=D_2=RR$, and $D_3=FCFS$),
 (— — $D_1=D_2=LCFS - PR$, and $D_3=FCFS$)

5. Selfish Scheduling Algorithms

In the previous chapters we have described and analyzed scheduling algorithms which provide : no discrimination with respect to service time (batch processing); linear discrimination (RR and LCFS - PR); discrimination in favour of short jobs (FB); and a family of multilevel scheduling algorithms. Now we are going to generalize all of these, creating a continuum of "selfish" scheduling algorithms. Those systems were first studied by Kleinrock, (KLEI 70a) the principle being that all customers in the computer system were divided into two groups: those in a "queue box" waiting for service; and those in a "service box" given service according to some scheduling algorithm (see fig. 5.1.) The material here is mainly expository and the reader wanting a more detailed treatment is referred to the excellent book by Kleinrock (KLEI 75b).



Fig. 5.1

An arrival always enters the queue box, where he is given a priority (a numerical value) that increases from zero at a positive rate α ; i.e. when he has spent t seconds in the queue box his priority is αt ; similarly, whenever a job is in the service box, his priority increases at a positive rate β . All customers possess the same parameters - α and β - and we restrict our attention to the case $0 \leq \beta \leq \alpha$.

A customer will pass from the queue box to the service box when his priority reaches a value that equals the priority in the service box (note that all customers in the service box must have the same priority, which grows at a rate β).

If the service box becomes completely empty, then that customer (if any) with the highest priority in the queue box will be transferred into the service box. A customer that arrives to an empty system will immediately pass into the service box, and his priority then grows from zero at a rate β . Since the customers in the service box are attempting to "run away" with the service facility, the algorithms are called "selfish" scheduling algorithms.

In order to distinguish between the scheduling algorithms we will call the algorithm employed in the service box the "raw" scheduling algorithm. Kleinrock shows that the transform of the conditional waiting time density in an arbitrary SSA system (selfish scheduling algorithms) can be written as

$$W^*(s|x) = \left(\frac{1-\rho}{1-\rho'}\right) \left(\frac{s-\lambda'+\lambda'B^*(s)}{s-\lambda'+\lambda B^*(s)}\right) \hat{W}_{\lambda'}^*(s|x) \quad (5.1)$$

where

$$\lambda' = \lambda(1-\beta/\alpha)$$

$$\rho' = \lambda' \bar{x}$$

and where $\hat{W}_{\lambda'}^*(s|x)$ is the transform of the conditional waiting time density in a system with the raw scheduling algorithm, the system being an M/G/1 system with arrival rate λ' .

The mean conditional response time for any SSA system is easily obtained from Eq. (5.1)

$$T(x) = \frac{\lambda x^2}{2(1-\rho)} - \frac{\lambda' x^2}{2(1-\rho')} + \hat{T}_{\lambda'}(x) \quad (5.2)$$

where $\hat{T}_{\lambda'}(x)$ is the mean conditional response time for the raw scheduling algorithm with input rate λ' .

I shall now demonstrate the behaviour of these SSA systems by means of some examples. Suppose that the raw scheduling algorithm is RR. We have an SRR (selfish round robin) system, and we immediately know that

$$\hat{T}_{\lambda}(x) = \frac{1}{1-\rho} x \quad (5.3)$$

If we put this into equation (5.2) we can easily see that

$$T(x) = \frac{\beta/\alpha}{1-\rho} T_{FCFS}(x) + (1 - \frac{\beta/\alpha}{1-\rho}) T_{RR}(x) \quad (5.4)$$

where

$$T_{FCFS}(x) = \frac{\lambda x^2}{2(1-\rho)} + x \quad (5.5)$$

and

$$T_{RR}(x) = \frac{x}{1-\rho} \quad (5.6)$$

Equation (5.4) exposes the linear mixing of the SRR system. We note especially that

$$T(x) = T_{FCFS}(x) \quad (5.7)$$

if $0 < \beta = \alpha$, which implies that the system behaves as a pure FCFS system - which it should, since those customers that are in the queue box can never catch the one that is in the service box, and since therefore the oldest arrival will be served to completion. If, on the other hand, $\beta/\alpha = 0$, then customers will enter with - and always maintain - a priority of zero, and therefore all the customers will be in the service box and will be served under the raw scheduling algorithm, namely RR. In figure 5:2 we give an example of the mean conditional waiting time in the SRR system for the case of exponential service time. N.B. The point of intersection is equal to $\bar{x}$.

As a second example, consider the SFB system, i.e. the raw scheduling algorithm FB. This gives us

$$T(x) = \frac{\lambda \bar{x}^2}{2(1-\rho)} - \frac{\lambda' \bar{x}^2}{2(1-\rho')} + \frac{\lambda' \bar{x}^2}{2(1-\rho'_x)} + \frac{x}{1-\rho'_x} \quad (5.8)$$

where

$$\rho'_x = \rho_x(1-\beta/\alpha)$$

For the same system as in figure 5:2, the SFB system produces figure 5:3.

Thus it is possible to take any solved raw scheduling algorithm and display the result for an SSA system. We are able to provide a continuum of response curves that depend upon the parameter β/α . This generality will cost almost nothing in implementation, since the only thing to be done is to count at a rate α for units in the queue box, and at a rate β for units in the service box.

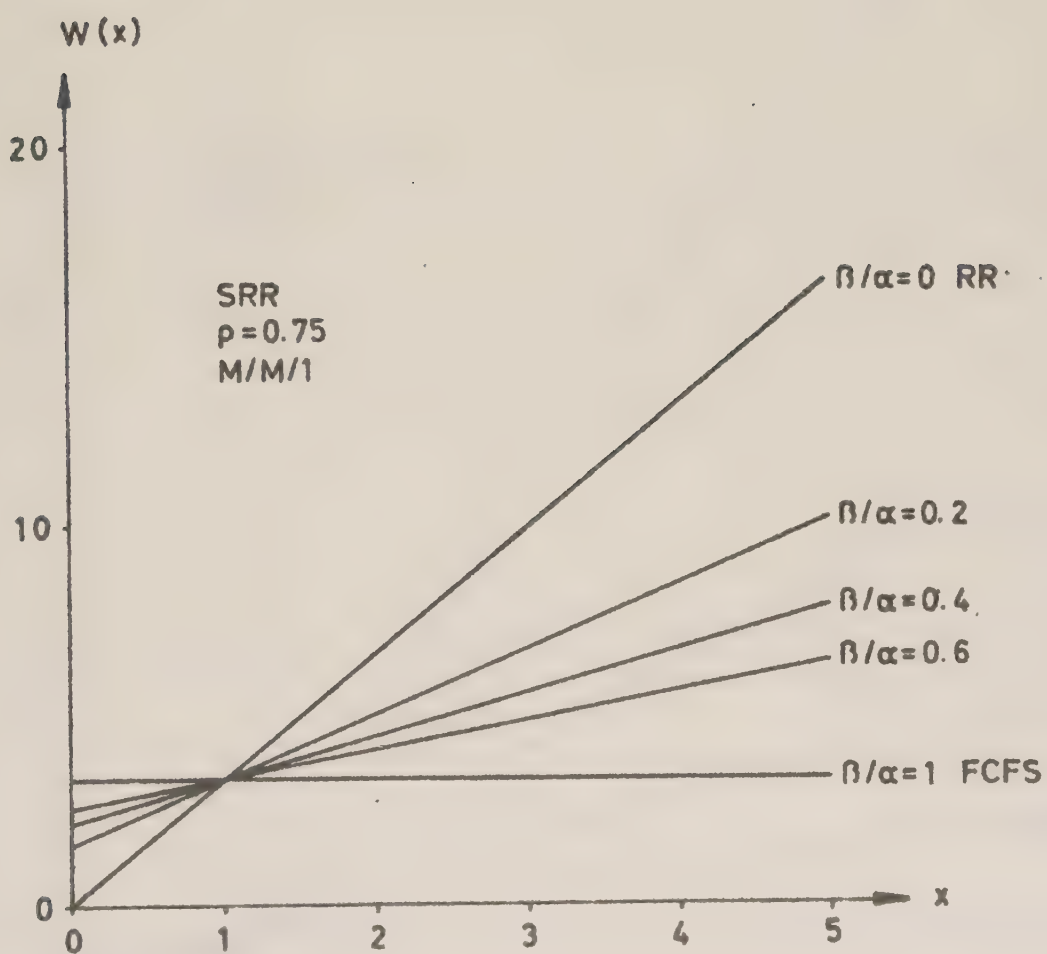


Fig. 5:2. Average waiting time functions for the SRR system with exponential service distribution. $\lambda = 0.75$, $\bar{x} = 1.0$.

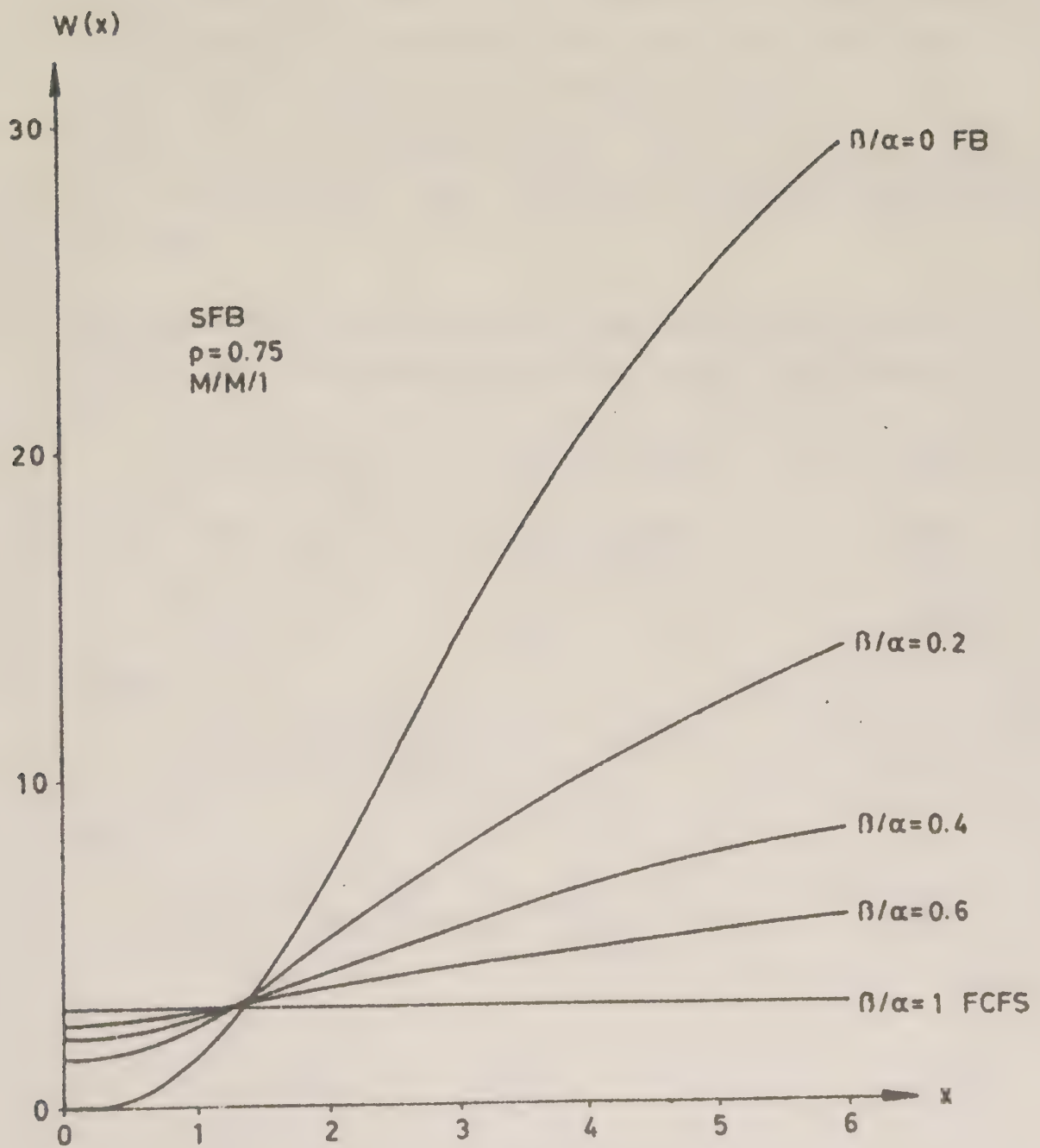


Fig. 5:3. Average waiting time functions for the SFB system with exponential service distribution. $\lambda = 0.75$, $\bar{x} = 1.0$.

6. Bounds, Inequalities and a Conservation Law

If one plots the mean waiting time conditioned on service requirement i.e. $W(x)$ as a function of x , for different scheduling disciplines one finds that the curves have a tendency to fall in a given region of the $(W(x), x)$ plane. Very quickly we find a pattern where by we could sketch out the boundaries within which the curves would fall. In (KLEI 71a) tight upper and lower bounds were established for $W(x)$.

We choose here just to report them rather than giving a full proof. Kleinrock shows that any $W(x)$ must obey

$$W(x) \geq W_L(x) \quad (\text{lower bound}) \quad (6.1)$$

$$W(x) \leq W_U(x) \quad (\text{upper bound}) \quad (6.2)$$

$$\frac{dW}{dx} \geq 0 \quad (6.3)$$

where

$$W_L(x) = \frac{\lambda x^2}{2(1-\rho_x)} \quad (6.4)$$

and

$$W_U(x) = \frac{\lambda x^2}{2(1-\rho_x)(1-\rho)} + \frac{x\rho_x}{1-\rho_x} \quad (6.5)$$

Note that

$$W_U(0) = W_L(\infty) = \frac{\lambda x^2}{2(1-\rho)} \quad (6.6)$$

which is exactly mean unfinished work in an M/G/1 system.

Furthermore

$$\lim_{x \rightarrow \infty} \frac{d}{dx} W_U(x) = \frac{\rho}{1-\rho} \quad (6.7)$$

which implies that for large x the upper bound has the same slope as the RR-algorithm. In figure 6:1 we show a collection of response curves with the tight upper and lower bounds superimposed.

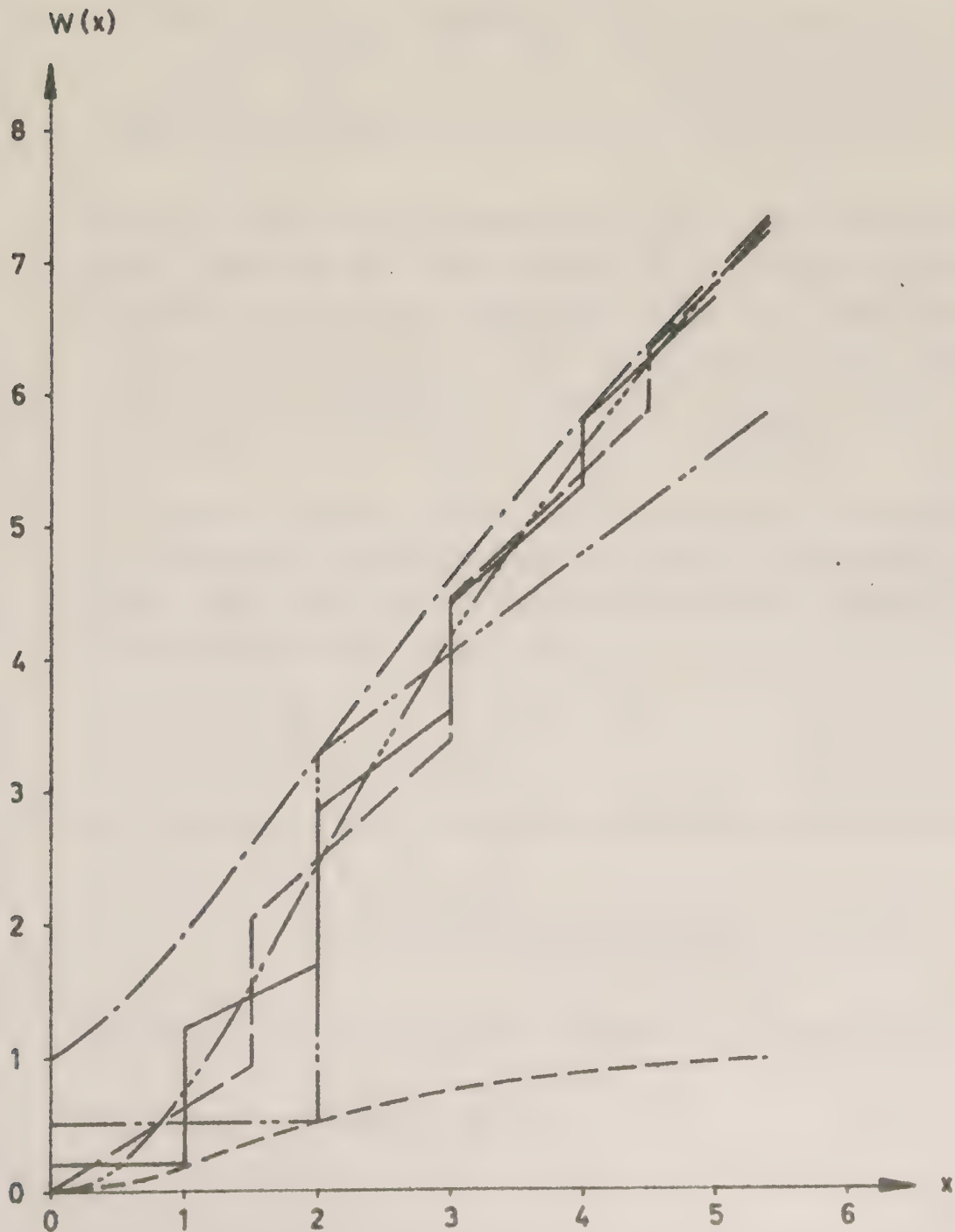


Fig. 6:1. A collection of response curves with the lower and upper bounds superimposed.

At $x=0$, the upper bound and the FCFS-response curve start at the same point, below we are going to show why that is so. The upper bound approaches the FB-curve asymptotically as x goes to infinity, therefore the worst treatment a customer with long service time can get is to be served under an FB-discipline. The lower bound starts at zero (as do FB, RR and LCFS-PR), and it increases less rapidly with x than the upper bound. It approaches the FCFS curve as x gets very large. We see that FCFS is the worst possible for extremely short jobs and is the best possible for very long jobs. The opposite is true for the FB-discipline.

Next we establish a conservation law, that $W(x)$ has to obey. This law was first proven in (KLEI 71a) and recently another proof has been given by (ODON 74). The basic underlying idea in the conservation law is that if we give some jobs preferential treatment in front of others we are forced to give other jobs worse treatment.

In an M/G/1 system the mean unfinished work is independent of the scheduling discipline, as long as the discipline falls into the category that was defined in chapter 2. We know that the mean work $\bar{U}$ is

$$\bar{U} = \frac{\lambda \bar{x}^2}{2(1-\rho)} = W_{FCFS} \quad (6.8)$$

On the other hand, we may also calculate this mean work as

$$\bar{U} = \int_0^{\infty} n(x) E\{\text{remaining service time for a job with attained service } x\} dx \quad (6.9)$$

where $n(x)$ is the attained service time density and

$$n(x) = \lambda(1-B(x)) \frac{dT}{dx} \quad (6.10)$$

We also know that

$$E\{\text{remaining service for a job with attained service } x\} = \int_x^{\infty} (y-x) \frac{dB(y)}{1-B(x)} \quad (6.11)$$

Putting this last expression into the integral for $\bar{U}$ it is found that

$$\int_0^{\infty} T(x)(1-B(x))dx = \frac{\bar{x}^2}{2(1-\rho)} \quad (6.12)$$

and

$$\int_0^{\infty} W(x)(1-B(x))dx = \frac{\rho \bar{x}^2}{2(1-\rho)} \quad (6.13)$$

The last expression follows since

$$\int_0^{\infty} x(1-B(x))dx = \frac{\bar{x}^2}{2} \quad (6.14)$$

We refer to equations (6.12) and (6.13) as Conservation Laws, since they are based on the conservation of average unfinished work in the system. The implication that can be drawn from the conservation law is that if we have given a $W(x)$, as a result of some scheduling algorithm, and then changed the algorithm so as to reduce $W(x)$ over some interval $(0, x_0)$, then the conservation law would require that the new $W(x)$ would be above the old value for some range above x_0 . This is due to the fact that the weighting factor, $1-B(x)$ is a decreasing function of x . In chapter 8 we will have the opportunity to return to the conservation law and use it in an optimization problem.

7. The Variance of Conditional Response Time.

In Chapter 3 we found that the mean conditional response time was not a good indicator of system performance. We showed f.i. that the mean conditional response time was the same for RR- and LCFS-PR- scheduling. Naturally we would like to have a better measure of system performance. The ultimate goal would be to determine the conditional distribution function of the response time, i.e.

$$S(y|x) = P\{\bar{y} \leq y \mid \bar{x}=x\} \quad (7.1)$$

and/or the Laplace transform of the associated density function:

$$S^*(s|x) = \int_0^{\infty} e^{-sy} dS(y|x) \quad (7.2)$$

Often, however, $S(y|x)$ and/or $S^*(s|x)$ are very hard to determine and the information they carry about system performance is not easily extracted, due to the complexity of the formulae. Naturally we can always extract information by using numerical computations. But these computations are not, in most cases, trivial, since numerical instabilities frequently occur. On the other hand, we do not always want the full information as given by $S(y|x)$ and/or $S^*(s|x)$, but are instead quite satisfied with ob-

taining higher moments of the conditional response time.

So in this chapter we will try to determine the variance of conditional response time.

$$\sigma^2(x) = \text{Var}\{\tilde{y}|\tilde{x}=x\} \quad (7.3)$$

The variance can, as usual, be computed from $S^*(s|x)$, since

$$E\{\tilde{y}^2|\tilde{x}=x\} = \frac{d^2 S^*(s|x)}{ds^2} \Big|_{s=0} \quad (7.4)$$

In the sequel we will study $\sigma^2(x)$ for the scheduling disciplines FCFS, FB, LCFS-PR, and RR, when the queueing system under study is of the type M/G/1. At least we will try to push the analysis as far as possible for general service time distributions. In some cases, however, we have to restrict the allowed service time distribution somewhat. Hopefully, the variance of conditional response time will be a good complementary indicator of system performance, to the mean conditional response time.

7.1 FCFS - Scheduling

If the scheduling discipline is FCFS the general theory for M/G/1 queues tells us that

$$S^*(s|x) = W^*(s) e^{-sx} \quad (7.5)$$

where

$$W^*(s) = \frac{s(1-\rho)}{s-\lambda+\lambda B^*(s)} \quad (7.6)$$

From equations (7.5) and (7.6) it is easily found that

$$\sigma^2(x) = \frac{\lambda \overline{x^3}}{3(1-\rho)} + \left(\frac{\lambda \overline{x^2}}{2(1-\rho)} \right)^2 \quad (7.7)$$

$\sigma^2(x)$ is independent of x , and it should, since FCFS does not discriminate with respect to service requirement.

7.2 FB - Scheduling.

In the case when scheduling discipline is FB, Schrage has shown that (SCHR 67)

$$S^*(s|x) = W_X^*(s+\lambda-\lambda G_X^*(s)) e^{-x(s+\lambda-\lambda G_X^*(s))} \quad (7.8)$$

where

$$G_X^*(s) = B_X^*(s+\lambda-\lambda G_X^*(s)) \quad (7.9)$$

and

$$W_X^*(s) = \frac{s(1-\rho_X)}{s-\lambda+\lambda B_X^*(s)} \quad (7.10)$$

Using equations (7.8) - (7.10) we find that

$$\sigma^2(x) = \frac{3W_x^2}{(1-\rho_x)^2} + \frac{\lambda x^3}{3(1-\rho_x)^3} + \frac{2W_x}{(1-\rho_x)^2} x \quad (7.11)$$

7.3 LCFS-PR- Scheduling.

The transform $S^*(s|x)$ can, under LCFS-PR scheduling, be determined by using a busy period analysis. A straight-forward analysis will lead to

$$S^*(s|x) = \exp\{-sx - \lambda x + \lambda x G^*(s)\} \quad (7.12)$$

where $G^*(s)$ is the transform of the density function for the length of a busy period in an M/G/1 system. $G^*(s)$ can be found through the functional equation.

$$G^*(s) = B^*(s + \lambda - \lambda G^*(s)) \quad (7.13)$$

$\sigma^2(x)$ is then readily computed and the result is

$$\sigma^2(x) = \frac{\lambda x^2}{(1-\rho)^3} x \quad (7.14)$$

Let us try to invert equation (7.12). We will demonstrate that even for the simple system M/M/1, the results are not very tractable.

We have

$$B^*(s) = \frac{\mu}{s + \mu} \quad (7:15)$$

which we use in equation (7:13) to obtain

$$G^*(s) = \frac{\mu}{s + \lambda - \lambda G^*(s) + \mu} \quad (7:16)$$

or

$$\lambda \{G^*(s)\}^2 - (s + \lambda + \mu)G^*(s) + \mu = 0 \quad (7:17)$$

Solving for $G^*(s)$ and restricting our solution to the case for which $|G^*(s)| \leq 1$ for $\text{Re}(s) \geq 0$, gives

$$G^*(s) = \frac{s + \mu + \lambda - \sqrt{(s + \lambda + \mu)^2 - 4\lambda\mu}}{2\lambda} \quad (7:18)$$

The expression for $G^*(s)$ is put into equation (7:12), which reads

$$S^*(s|x) = \exp \left\{ -\frac{sx}{2} + \frac{x}{2} (\mu - \lambda) - \frac{x}{2} \sqrt{(\mu + \lambda + s)^2 - 4\lambda\mu} \right\} \quad (7:19)$$

This equation may be inverted by using transform tables (ERDE 54) to obtain the conditional probability density

function for the time in system, namely

$$\begin{aligned}
 s(y|x) &= e^{-\lambda x} e^{-(\mu+\lambda)(y-x)} \{ \delta(y-x) + \\
 &+ x \sqrt{\lambda \mu} \sqrt{y^2 - yx} I_1 \{ \sqrt{4\lambda \mu (y^2 - yx)} \} \\
 &y \geq x
 \end{aligned}
 \tag{7.20}$$

where I_1 is the modified Bessel function of the first kind of order one.

Equation (7:20) is certainly not encouraging. It shows that even for the simple system, M/M/1, the result turns out to be quite complex, and our understanding of system performance is definitely not enhanced by such an equation.

Let us take another example, the system M/E₂/1, and examine how well we can do - although we might expect that our success will be limited. Using the expression for $B^*(s)$ in the functional equation (7:13) we get

$$G^*(s) = \left(\frac{2\mu}{s + \lambda - \lambda G^*(s) + 2\mu} \right)^2
 \tag{7:21}$$

which leads to the cubic equation

$$\lambda^2 (G^*(s))^3 - (G^*(s))^2 2\lambda(s + \lambda + 2\mu) + G^*(s) (s + \lambda + 2\mu)^2 - 4\mu^2 = 0
 \tag{7:22}$$

This last equation is not easily solved and so, at this point, we stop, in our attempt to invert $S^*(s|x)$.

These two examples show that instead of looking for explicit formulas for $s(y|x)$ we should try to find other means of extracting information from equations (7.12) and (7.13). One possible way would be to apply a numerical inversion technique to these equations.

7.4 RR - Scheduling.

The RR scheduling discipline produced a very fair and elegant expression for the mean conditional response time (see chapter 3). The variance of conditional response time is somewhat harder to compute. We know of only one system, the M/M/1 queue, for which the variance has been found (COFF 70). In this paper by Coffman et al. a solution of $S^*(s|x)$ was presented when

$$B(x) = 1 - e^{-\mu x} \quad x \geq 0 \quad (7.23)$$

From $S^*(s|x)$ the variance can be computed and it reads

$$\sigma^2(x) = \frac{2\rho x}{\mu(1-\rho)^3} - \frac{2\rho}{\mu^2(1-\rho)^4} (1 - e^{-\mu(1-\rho)x}) \quad (7.24)$$

Note that there is a misprint in the article by Coffman et. al. in their equation for $\sigma^2(x)$. Equation (7.24) above

gives the correct formula for $\sigma^2(x)$.

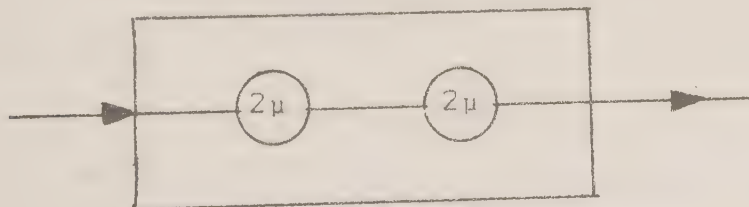
Unfortunately no solution for $\sigma^2(x)$ has been found when the service time distribution is general.

The variance can, however, be determined for some other service time distributions, and below the analysis for the $M/E_2/1$ system is going to be presented. The method we will use is, in fact, a generalization of the method used in (COFF 70). Furthermore, the method is applicable to all systems with service time distributions such that $B^*(s)$ is a rational function of s .

Let us now analyse the $M/E_2/1$ queue. For this system the service time distribution is Erlang-2, which means that

$$b(x) = 2\mu (2\mu x) e^{-2\mu x} \quad x \geq 0 \quad (7:25)$$

An Erlang-2 distribution can be looked upon as consisting of two independent exponential stages (COX 55). Conceptually, the service mechanism in the $M/E_2/1$ system can therefore be depicted as in fig 7:1



Service mechanism for an $M/E_2/1$ queue

Figure 7:1

When the queueing discipline is RR, it is known (MUNT 73) that the state probabilities $p(n_1, n_2)$ defined by

$$p(n_1, n_2) \triangleq P\{n_1 \text{ customers in stage 1,} \\ n_2 \text{ customers in stage 2.}\} \quad (7.26)$$

can be expressed as

$$p(n_1, n_2) = (1-\rho) \binom{n_1+n_2}{n_1} \left(\frac{\lambda}{2\mu}\right)^{n_1} \left(\frac{\lambda}{2\mu}\right)^{n_2} \quad (7.27)$$

where $\rho \triangleq \lambda/\mu$

Now let $w_{n_1, n_2}(x)$ denote a random variable that corresponds to the waiting time experienced by a customer with service requirement x , and who, upon arrival, finds n_1 customers in stage 1, and n_2 customers in stage 2. Then $w_{n_1, n_2}(x) + x$ becomes the response time for this customer. Our objective is to find the variance of conditional response time, and this must be equivalent to finding the variance of the conditional waiting time. We define

$$W_{n_1, n_2}^*(x, s) \triangleq E\{e^{-s w_{n_1, n_2}(x)}\} \quad (7.28)$$

If we were able to find $W_{n_1, n_2}^*(x, s)$, we might easily

determine $W^*(x,s)$ - the Laplace transform of the density function for the conditional waiting time, and

$$W^*(x,s) = \sum_{n_1} \sum_{n_2} W_{n_1, n_2}^*(x,s) p(n_1, n_2) \quad (7.29)$$

from equation (7.29), the mean and variance of the conditional waiting time can be found.

A relation for $W_{n_1, n_2}^*(x,s)$ can be derived in the following way. Let Δx be a differential element of (service) time.

The tagged customer must spend $(n_1 + n_2 + 1) \Delta x$ seconds in the system before he has received Δx seconds of service.

Neglecting terms of the order $\sigma(\Delta x)$, the probability of an arrival during this time interval is

$\lambda (n_1 + n_2 + 1) \Delta x$, the probability of a customer leaving stage 1 and entering stage 2 is

$2\mu (n_1 + n_2 + 1) \Delta x n_1 / (n_1 + n_2 + 1) = 2\mu n_1 \Delta x$,

the probability of a departure is $2\mu n_2 \Delta x$, and the

probability of no change is $1 - (\lambda(n_1 + n_2 + 1) + 2\mu(n_1 + n_2)) \Delta x$.

Hence, again neglecting terms of the order $\sigma(\Delta x)$, we may write

$$E\{e^{-sw_{n_1, n_2}}(x + \Delta x)\} = (1 - (\lambda(n_1 + n_2 + 1) + 2\mu(n_1 + n_2)) \Delta x$$

$$\cdot e^{-s\Delta x(n_1 + n_2)} E\{e^{-sw_{n_1, n_2}}(x)\} + \quad (7.30)$$

$$+ \lambda(n_1+n_2+1) \Delta x \ E\{ e^{-sW_{n_1+1, n_2}}(\dot{x}) \} + \quad (7.30)$$

$$+ 2\mu n_1 \Delta x \ E\{ e^{-sW_{n_1-1, n_2+1}}(x) \} +$$

$$+ 2\mu n_2 \Delta x \ E\{ e^{-sW_{n_1, n_2-1}}(x) \}$$

Rearranging, taking the limit $\Delta x \rightarrow 0$ and using $W_{n_1, n_2}^*(x, s)$

we obtain

$$\begin{aligned} \frac{\partial W_{n_1 n_2}^*(x, s)}{\partial x} &= -(\lambda(n_1+n_2+1) + (s+2\mu)(n_1+n_2))W_{n_1 n_2}^*(x, s) \\ &+ \lambda(n_1+n_2+1) W_{n_1+1, n_2}^*(x, s) + 2\mu n_1 W_{n_1-1, n_2+1}^*(x, s) \\ &+ 2\mu n_2 W_{n_1, n_2-1}^*(x, s) \end{aligned} \quad (7.31)$$

In order to gain more information from (7.31) it is convenient to introduce a generating function:

$$f = f(x, s, z, y) = \sum_{n_2=0}^{\infty} \sum_{n_1=0}^{\infty} W_{n_1, n_2}^*(x, s) \binom{n_1+n_2}{n_1} z^{n_1} y^{n_2} \quad (7.32)$$

Applying equation (7.32) to equation (7.31), and introducing the obvious boundary conditions, will result in

$$\begin{aligned} \frac{\partial f}{\partial x} - (z(2\mu y - s - \lambda - 2\mu) + \lambda) \frac{\partial f}{\partial z} - \\ -(y(2\mu y - s - \lambda - 2\mu) + 2\mu z) \frac{\partial f}{\partial y} = (2\mu y - \lambda) f \end{aligned} \quad (7.33)$$

- a partial differential equation in x , y and z . The equation can probably be solved, but we are, however, interested only in determining the variance of conditional response time, and because of this we choose to attack equation (7.33) in another way. First of all, let us determine the value of $f(x,s,z,y)$ for some combinations of the variables x,s,z and y .

If we put $s=0$ we find from equation (7.32) that

$$f(x,0,z,y) = \frac{1}{1-z-y} \quad (7.34)$$

Furthermore, if we put $z=y=\rho/2$ in (7.34) we get

$$f(x,0,\rho/2,\rho/2) = \frac{1}{1-\rho} \quad (7.35)$$

We can also see from equations (7.28) and (7.32) that, if $x=0$ which implies $w_{n_1,n_2}(x) = 0$, then

$$f(0,s,z,y) = \frac{1}{1-z-y} \quad (7.36)$$

More interesting is perhaps that by using equations (7.27), (7.29), and (7.32) we find

$$W^*(x,s) = (1-\rho)f(x,s,\rho/2,\rho/2) \quad (7.37)$$

Obviously this is the nice "trick", since our selection of such a generating function as $f(x,s,z,y)$ allows us to compute the mean and variance of conditional response time by computing derivatives with respect to s of $f(x,s,z,y)$.

Let us rewrite the partial-differential equation (7.33) as

$$\frac{\partial f}{\partial x} - h(s,z,y) \frac{\partial f}{\partial z} - g(s,z,y) \frac{\partial f}{\partial y} = (2\mu y - \lambda)f \quad (7.38)$$

where

$$h(s,z,y) = z(2\mu y - s - \lambda - 2\mu) + \lambda \quad (7.39)$$

$$g(s,z,y) = y(2\mu y - s - \lambda - 2\mu) + 2\mu z \quad (7.40)$$

If we differentiate once with respect to s we get

$$\frac{\partial^2 f}{\partial s \partial x} - \frac{\partial h}{\partial s} \frac{\partial f}{\partial z} - h \frac{\partial^2 f}{\partial s \partial z} - \frac{\partial g}{\partial s} \frac{\partial f}{\partial y} - g \frac{\partial^2 f}{\partial s \partial y} = (2\mu y - \lambda) \frac{\partial f}{\partial s} \quad (7.41)$$

Equation (7.37) gives us that

$$W(x) = -(1-\rho) \left. \frac{\partial f}{\partial s} \right|_{\substack{s=0 \\ z=y=\rho/2}} \quad (7.42)$$

Plugging $s=0$ and $z=y=p/2$ into equation (7.41), and using (7.34) and (7.42), we arrive at

$$\frac{d}{dx} W(x) = \frac{\rho}{1-\rho} \quad (7.43)$$

and, after integration, at

$$W(x) = \frac{x\rho}{1-\rho} \quad (7.44)$$

since $W(0) = 0$

This is the wellknown formula for the mean conditional waiting time in an M/G/1 system that operates under RR scheduling. This result could be obtained differently and more easily, but it is nice to be able to confirm that we are heading in the right direction. Now, differentiation of (7.41) with respect to s , yields

$$\begin{aligned} \frac{\partial^3 f}{\partial s^2 \partial x} - \frac{\partial^2 h}{\partial s^2} \frac{\partial f}{\partial z} - 2 \frac{\partial h}{\partial s} \frac{\partial^2 f}{\partial s \partial z} - h \frac{\partial^3 f}{\partial s^2 \partial z} - \frac{\partial^2 g}{\partial s^2} \frac{\partial f}{\partial y} - \\ - 2 \frac{\partial g}{\partial s} \frac{\partial^2 f}{\partial s \partial y} - g \frac{\partial^3 f}{\partial s^2 \partial y} = (2\mu - \lambda) \frac{\partial^2 f}{\partial s^2} \end{aligned} \quad (7.45)$$

Let us introduce

$$\alpha(x) = \left. \frac{\partial^2 f}{\partial s^2} \right|_{\substack{s=0 \\ z=y=p/2}} \quad (7.46)$$

$$q(x) = \frac{\partial^2 f}{\partial s \partial z} \bigg|_{\substack{s=0 \\ z=y=\rho/2}} \quad (7.47)$$

and

$$p(x) = \frac{\partial^2 f}{\partial s \partial y} \bigg|_{\substack{s=0 \\ z=y=\rho/2}} \quad (7.48)$$

Put $s=0$ and $z=y=\rho/2$ into equation (7.45), and we get

$$\frac{d\alpha(x)}{dx} = -\rho(q(x) + p(x)) \quad (7.49)$$

Equations for $q(x)$ and $p(x)$, which can also be obtained from (7.41), are:

$$\frac{dq(x)}{dx} + 2\mu q(x) - 2\mu p(x) = -\frac{1+\rho}{(1-\rho)^3} \quad (7.50)$$

$$\frac{dp(x)}{dx} - (\lambda - 2\mu)p(x) - \lambda q(x) = -x \frac{2\mu\rho}{(1-\rho)^3} - \frac{1+\rho}{(1-\rho)^3} \quad (7.51)$$

The quantity we want to solve for is obviously $\alpha(x)$, since

$$E\{\tilde{w}^2 \mid \tilde{x} = x\} = (1-\rho) \alpha(x) \quad (7.52)$$

where $\tilde{w} = \tilde{y} - x$

The solution of $\alpha(x)$ is, however, easily found if we use Laplace transform technique. We know that $\alpha(0)=0$ and therefore

$$\int_0^{\infty} \alpha(x) e^{-sx} dx = - \frac{\rho}{s} (Q^*(s) + P^*(s)) \quad (7.53)$$

where

$$Q^*(s) = \int_0^{\infty} q(x) e^{-sx} dx \quad (7.54)$$

and

$$P^*(s) = \int_0^{\infty} p(x) e^{-sx} dx \quad (7.55)$$

Applying (7.54) and (7.55) to (7.50) and (7.51), and rearranging, will lead to

$$\begin{aligned} \frac{Q^*(s) + P^*(s)}{s} (s^2 - s(\lambda - 4\mu) + 4\mu^2 - 4\lambda\mu) = \\ = \frac{-2(1+\rho)}{(1-\rho)^3 s} + \frac{2\mu\rho^2 - 8\mu\rho - 6\mu}{s^2(1-\rho)^3} - \frac{8\mu^2\rho}{(1-\rho)^2 s^3} \end{aligned} \quad (7.56)$$

The variance of conditional waiting time (response time) can be written as

$$\sigma^2(x) = E\{\tilde{w}^2 \mid \tilde{x}=x\} - E^2\{\tilde{w} \mid \tilde{x}=x\} \quad (7.57)$$

Hence, it follows that

$$\int_0^{\infty} \sigma^2(x) e^{-sx} = \frac{6\mu\rho + 2\rho s}{(1-\rho)^2 s^2 (s^2 - s(\lambda - 4\mu) + 4\mu^2 - 4\mu\lambda)} \quad (7.58)$$

or if we prefer to work in the time domain

$$\begin{aligned} \sigma^2(x) = & \frac{3\rho x}{2\mu(1-\rho)^3} - \frac{6\mu\rho + 2\rho s_1}{(1-\rho)^2 s_1^2 (s_1 - s_2)} (1 - e^{s_1 x}) \\ & - \frac{6\mu\rho + 2\rho s_2}{(1-\rho)^2 s_2^2 (s_2 - s_1)} (1 - e^{s_2 x}) \end{aligned} \quad (7.59)$$

where s_1 and s_2 are roots to the equation

$$s^2 - s(\lambda - 4\mu) + 4\mu^2 - 4\mu\lambda = 0 \quad (7.60)$$

It is easy to show that if the system under study is stable, i.e. $\rho < 1$, then s_1 and s_2 are distinct and negative. The linear term of equation (7.59) can be written as

$$\frac{\lambda x^2}{(1-\rho)^3} x$$

This is quite surprising since it is the same kind of linear term as was found for the M/M/1 system. $\sigma^2(x)$

has also been found for the service time distributions H_2 and Erlang - 3. Below we list the transform of $\sigma^2(x)$ for the cases that have a known solution:

$$\int_0^{\infty} \sigma^2(x) e^{-sx} dx = \frac{\lambda x^2 \mu}{(1-\rho)^2 s^2 (s+\mu-\lambda)} \quad (7.62)$$

when $b(x) = \mu e^{-\mu x}$

$$\frac{\lambda x^2 (2\mu)^2 + 2\rho s}{(1-\rho)^2 s^2 (s-s_1) (s-s_2)} \quad (7.63)$$

when $b(x) = 2\mu (2\mu x) e^{-2\mu x}$

and s_1 and s_2 are roots to the equation

$$s^2 + s(4\mu - \lambda) + 4\mu^2(1-\rho) = 0 \quad (7.64)$$

$$\frac{\lambda x^2 (3\mu)^3 + 16\lambda s + 2\rho s^2}{(1-\rho)^2 s^2 (s-s_1) (s-s_2) (s-s_3)} \quad (7.65)$$

$$\text{when } b(x) = 3\mu \frac{(3\mu x)^2}{2!} e^{-3\mu x}$$

where s_1 , s_2 , and s_3 are roots to the equation

$$s^3 + s^2(9\mu - \lambda) + s(27\mu^2 - 9\lambda\mu) + 27\mu^3(1-\rho) = 0 \quad (7.66)$$

$$\frac{\lambda x^2 \mu_1 \mu_2 + 2\rho s}{(1-\rho)^2 s^2 (s-s_1) (s-s_2)} \quad (7.67)$$

when $b(x) = \alpha \mu_1 e^{-\mu_1 x} + (1-\alpha) \mu_2 e^{-\mu_2 x}$

and s_1 and s_2 are roots to the equation

$$s^2 + s(\mu_1 + \mu_2 - \lambda) + \mu_1 \mu_2 (1-\rho) = 0 \quad (7.68)$$

The zeroes of the denominators in equations (7.62), (7.73), (7.65) and (7.67) can be determined through

$$s(s - \lambda + \lambda B^*(s)) = 0 \quad (7.69)$$

which might perhaps lead us to suspect that there exists some deeper generality in the formulae^{*}. This generality, however, has not yet been established. Furthermore, if we look at the linear term in the expressions for $\sigma^2(x)$, we will find its form to be

$$\frac{\lambda x^2}{(1-\rho)^3} x \quad (7.70)$$

* It is, however, possible to make a conjecture about the general formulae for the Laplace transform of $\sigma^2(x)$. Equations (7.62), (7.63), (7.65), and (7.67) lead us to the conjecture that, when the service time distribution is general the Laplace

which implies that the behaviour of $\sigma^2(x)$ for large x is such as described in equation (7.70).

Let us make an attempt to determine the variance of conditional response time for a job whose service requirement (x) is very large.

Let us introduce the notation

$\tilde{y}(x) \triangleq$ the response time for the tagged job
that requires x seconds service. (7.71)

When x is large we may express $\tilde{y}(x)$ as

$$\tilde{y}(x) = \tau + x + \sum_{i=1}^{N(\tilde{y}(x))} x_i \quad (7.72)$$

where τ is a stochastic variable that represents the remaining work in the system, when the tagged job arrives. All of that work has to be processed while the tagged job is in the system.

$N(\tilde{y}(x))$ is the number of customers that arrive to the system while the tagged job is there. These customers all have to be processed before the tagged job can leave the system.

As before, x_i is the stochastic variable that corresponds to the service time requirement of the i^{th} customer.

transform of $\sigma^2(x)$ should be

$$\int_0^{\infty} \sigma^2(x) e^{-sx} dx = \frac{2\rho}{(1-\rho)^2 s^3} \frac{1 - \hat{B}^*(s)}{1 - \rho \hat{B}^*(s)}$$

where

$$\hat{B}^*(s) = 1 - B^*(s)$$

Now, with a tongue in cheek derivation,

$$\text{Var}\{\hat{y}(x)\} = \text{Var}\{\tau\} + \text{Var}\left\{\sum_1^{N(\hat{y}(x))} x_i\right\} \quad (7.73)$$

and where the second term of the right hand side has to be dealt with in a special way.

$$\begin{aligned} \text{Var}\left\{\sum_1^{N(\hat{y}(x))} x_i\right\} &= E\left\{\left(\sum_1^{N(\hat{y}(x))} x_i\right)^2\right\} - \\ &- E^2\left\{\sum_1^{N(\hat{y}(x))} x_i\right\} \end{aligned} \quad (7.74)$$

Now let us assume that $\hat{y}(x) = y$, i.e. condition on $\hat{y}(x)$, and let

$$N(\hat{y}(x) \mid \hat{y}(x) = y) = N \quad (7.75)$$

Then we will want to compute

$$E\left\{\left(\sum_1^N x_i\right)^2\right\} \quad (7.76)$$

and

$$E\left\{\sum_1^N x_i\right\} \quad (7.77)$$

If N were independent of x_{N+1}, x_{N+2} , we could apply Wald's equation (FELL 66). This is unfortunately not true, as pointed out in many textbooks, e.g. (ROSS 70), since the event

$$\{N(t) = n\} \text{ implies } \\ \text{that } x_1 + x_2 + \dots + x_n \leq t \text{ and} \\ x_1 + x_2 + \dots + x_n + x_{n+1} > t$$

and thus the event $\{N(t) = n\}$ is not independent of x_{n+1} . If we use common terminology this means that

$N(t)$ is not a stopping time. However, it is easy to show that $N(t) + 1$ is a stopping time. This is true since

$$\{N(t) + 1 = n\} \Leftrightarrow \{N(t) = n - 1\}$$

and thus the event $\{N(t) = n-1\}$ is independent of x_{n+1}, x_{n+2} , etc.

Therefore we form

$$E \left\{ \sum_{i=1}^{N+1} x_i \right\} \tag{7.78}$$

and now it is possible to apply Wald's equation, which gives

$$E \left\{ \sum_{i=1}^{N+1} x_i \right\} = \bar{x} E(N+1) \quad (7.79)$$

The next step will be to study

$$E \left\{ \left(\sum_{i=1}^{N+1} x_i \right)^2 \right\} \quad (7.80)$$

and this can be written as

$$\begin{aligned} E \left\{ \left(\sum_{i=1}^{N+1} x_i \right)^2 \right\} &= E(N+1) \text{Var} \{x_i\} - E\{(N+1)^2\} \bar{x}^2 \\ &+ 2 \bar{x} E \left\{ (N+1) \sum_{i=1}^{N+1} x_i \right\} \end{aligned} \quad (7.81)$$

Equation (7.81) is a generalization of Wald's equation, and we will take the opportunity here to prove this (LAVE 75).

Since this is a digression, the reader not interested in the proof of (7.81) may skip over it.

Let X_i , $i \geq 1$, be independent, identically distributed non-negative random variables, Let N be a positive integer - valued random variable such that N and X_{N+1} , X_{N+2} , - - - are mutually independent.

Then

$$\begin{aligned} E\left\{\left(\sum_{i=1}^N X_i\right)^2\right\} &= E(N) \text{Var}(X) - \\ &- E(N^2) (E(X))^2 + 2E(X) E\left(N \sum_{i=1}^N X_i\right) \end{aligned} \quad (7.82)$$

This is true, since, if we define

$$I_i(N) = \begin{cases} 1 & i \leq N \\ 0 & i > N \end{cases} \quad (7.83)$$

We can write

$$\begin{aligned} E\left\{\left(\sum_{i=1}^N X_i\right)^2\right\} &= E\left\{\left(\sum_{i=1}^{\infty} I_i(N) X_i\right)^2\right\} = \\ &= E\left\{\sum_{j=1}^{\infty} I_j(N) X_j \sum_{i=1}^{\infty} I_i(N) X_i\right\} \end{aligned} \quad (7.84)$$

However, $I_j(N)$ and X_j are independent and furthermore

$$I_j(N) I_i(N) = I_j(N) \text{ if } j \geq i \quad (7.85)$$

Thus (7.84) can be written as

$$\begin{aligned} &\sum_{j=1}^{\infty} E\{X_j^2 I_j(N)\} + \sum_{j=2}^{\infty} \sum_{i=1}^{j-1} E\{X_j I_j(N) X_i\} \\ &+ \sum_{j=1}^{\infty} \sum_{i=j+1}^{\infty} E\{X_j X_i I_i(N)\} = \end{aligned} \quad (7.86)$$

$$= E(X^2) \sum_{j=1}^{\infty} P(N > j) + 2E(X) \sum_{j=1}^{\infty} \sum_{i=j+1}^{\infty} E(X_j I_i(N))$$

$$= E(X^2) E(N) + 2E(X) E\left\{ \sum_{j=1}^{\infty} X_j \sum_{i=j+1}^{\infty} I_i(N) \right\}$$

Observe that $\sum_{i=j+1}^{\infty} I_i(N) = \max(N-j, 0)$.

Therefore

$$\begin{aligned} E\left\{ \sum_{j=1}^{\infty} X_j \sum_{i=j+1}^{\infty} I_i(N) \right\} &= E\left\{ \sum_{j=1}^N X_j (N-j) \right\} \\ &= E\left\{ N \sum_{j=1}^N X_j \right\} - E\left\{ \sum_{j=1}^N j X_j \right\} \end{aligned} \quad (7.67)$$

where

$$\begin{aligned} E\left\{ \sum_{j=1}^N j X_j \right\} &= E\left\{ \sum_{j=1}^{\infty} j \psi_j(N) X_j \right\} \\ &= E(X) \sum_{j=1}^{\infty} j P(N \geq j) \\ &= E(X) (E(N^2) + E(N))/2 \end{aligned} \quad (7.88)$$

Equation (7.82) follows from (7.86) - (7.88), thus completing the proof.

Now, using equation (7.81) and Wald's theorem, as well as some other manipulations, it is found that

$$\begin{aligned} \text{Var}\{ \sum_{i=1}^{N(\tilde{y}(x))} x_i \} &= \overline{\lambda x^2} E\{\tilde{y}(x)\} - \rho^2 \text{Var}\{\tilde{y}(x)\} \\ &\quad + 2\rho \text{Var}\{\tilde{y}(x)\} \end{aligned} \quad (7.89)$$

From equations (7.73) and (7.89) we see that

$$\text{Var}\{\tilde{y}(x)\} (1-\rho)^2 \sim \overline{\lambda x^2} E\{\tilde{y}(x)\} \quad (7.90)$$

- the $\sim$ sign meaning that the right hand side contains more terms than is shown in equation (7.90) the terms not spelled out, however, are independent of x .

Hence for large x

$$\text{Var}\{\tilde{y}(x)\} \approx \frac{\overline{\lambda x^2}}{(1-\rho)^3} x \quad (7.91)$$

which shows that for any service time distribution, the conditional variance of the response time (waiting time) will be linear in x , for large x .

When x is very small the variance of conditional response time can also be determined for an M/G/1 system that operates under RR-scheduling. We know that the state-probabilities p_k can be expressed as

$$p_k = (1-\rho) \rho^k \quad k \geq 0 \quad (7.92)$$

Thus we can form

$$E\{e^{-sw(x)}\} = \sum_{k=0}^{\infty} E\{e^{-sw(x)} \mid k\} (1-\rho) \rho^k \quad (7.93)$$

For very small x it must be that

$$E\{e^{-sw(x)} \mid k\} = (e^{-sx})^k \quad (7.94)$$

and using (7.94) in (7.93) we get

$$E\{e^{-sw(x)}\} = (1-\rho) / (1-\rho e^{-sx}) \quad (7.95)$$

From (7.95) it is easily found that

$$\sigma^2(x) = \frac{\rho x^2}{(1-\rho)^2} \quad (7.96)$$

Note that (7.96) is only valid for x being very small.

This concludes our investigation of the variance of conditional waiting time under RR scheduling.

7.5 Did We Get a Better Measure of System Performance?

Did the variance of conditional response time provide us with a better measure of system performance? At least it gave us one more parameter that can be used if we want to compare scheduling algorithms. As stated above, the conditional mean response time $T(x)$ was not enough when we wanted to compare RR- and LCFS-PR- scheduling. The variance of conditional response time, however, clearly makes it possible to distinguish between the two scheduling disciplines. In those cases where we did obtain an explicit expression for $\sigma^2(x)$, when the scheduling algorithm was RR, we found that (equations (7.14, 7.24, 7.59, 7.65 and 7.67))

$$\sigma_{RR}^2(x) \leq \sigma_{LCFS-PR}^2(x) \quad (7.97)$$

In order to illustrate the behaviour of $\sigma^2(x)$ for different scheduling algorithms, some numerical examples have been computed. Figures 7:2,3,4 show $\sigma^2(x)$ as a function of the load ρ with x as a parameter for the systems $M/M/1$, $M/E_2/1$ and $M/H_2/1$. Figures 7:5,6,7 show the dependence between $\sigma^2(x)$ and x , when the load ρ is held constant. It is awfully difficult to make a statement about which algorithm is the best, since we don't know what criterion we will use. If we invent one criterion that says that

"the algorithm producing the smallest variance of conditional response time is the best" the diagrams will tell us that the RR- scheduling discipline is not at all a poor candidate. We are, however, painfully aware of the fact that this criterion is not sufficient and that, perhaps, we would like to invent one simultaneously containing $T(x)$ and $\sigma^2(x)$. We do not want to formulate such a criterion here, - rather, we leave that formidable task to the reader. This chapter certainly provides the "criteria inventor" with the necessary tools for such a task.

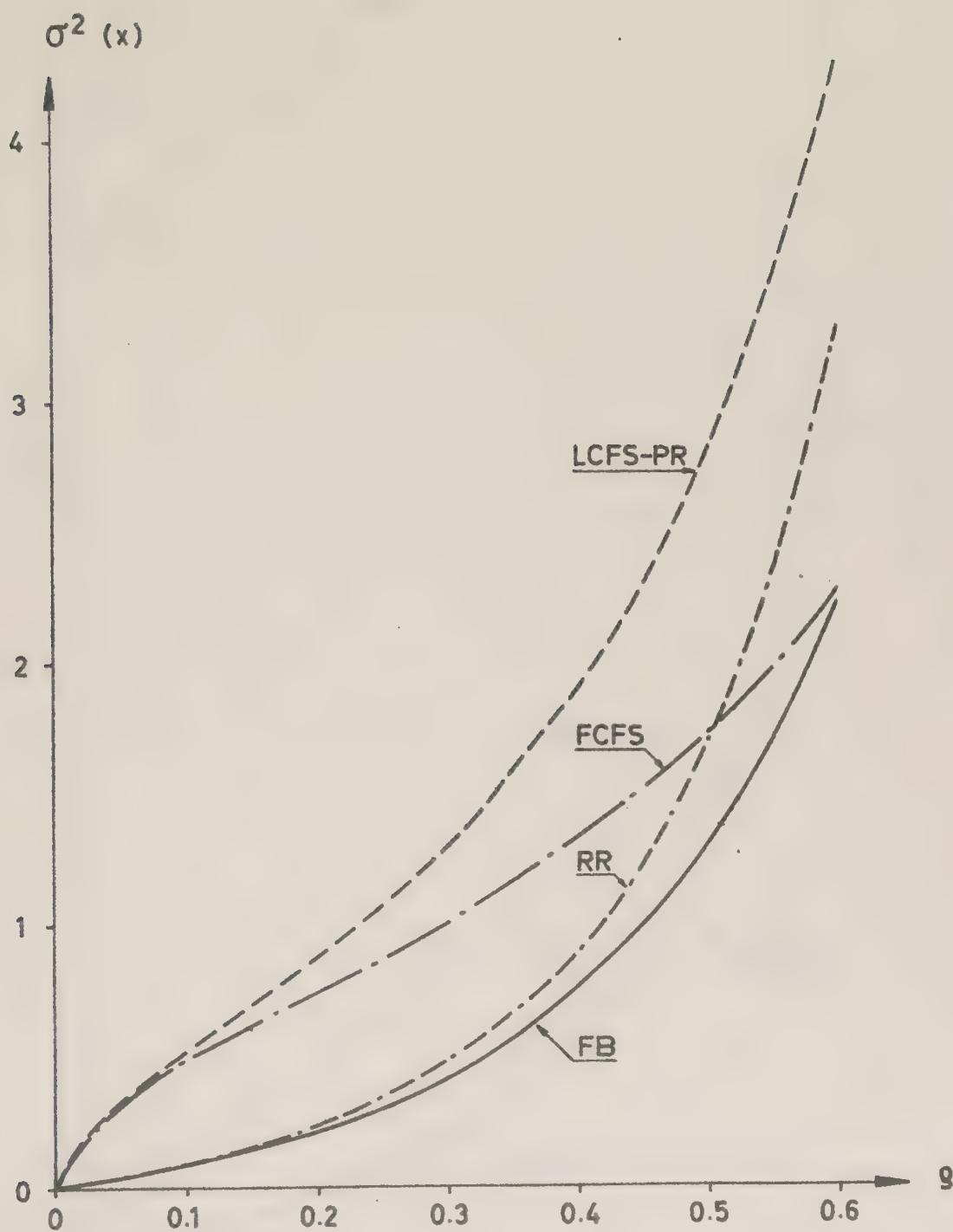


Fig. 7:2. Variance of conditional response time.

M/M/1. $x = 1$.

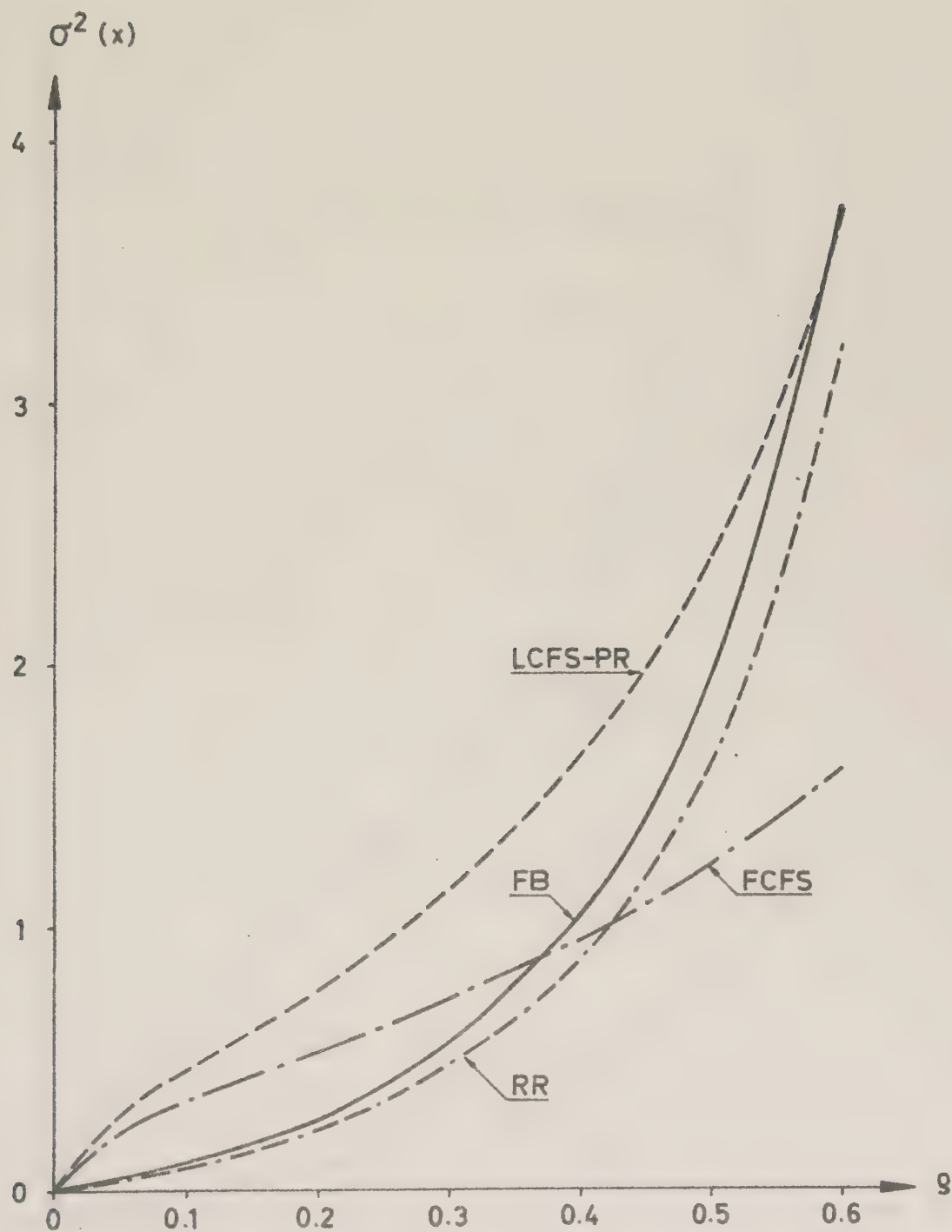


Fig. 7:3. Variance of conditional response time.

$M/E_2/1$. $x = 1$.

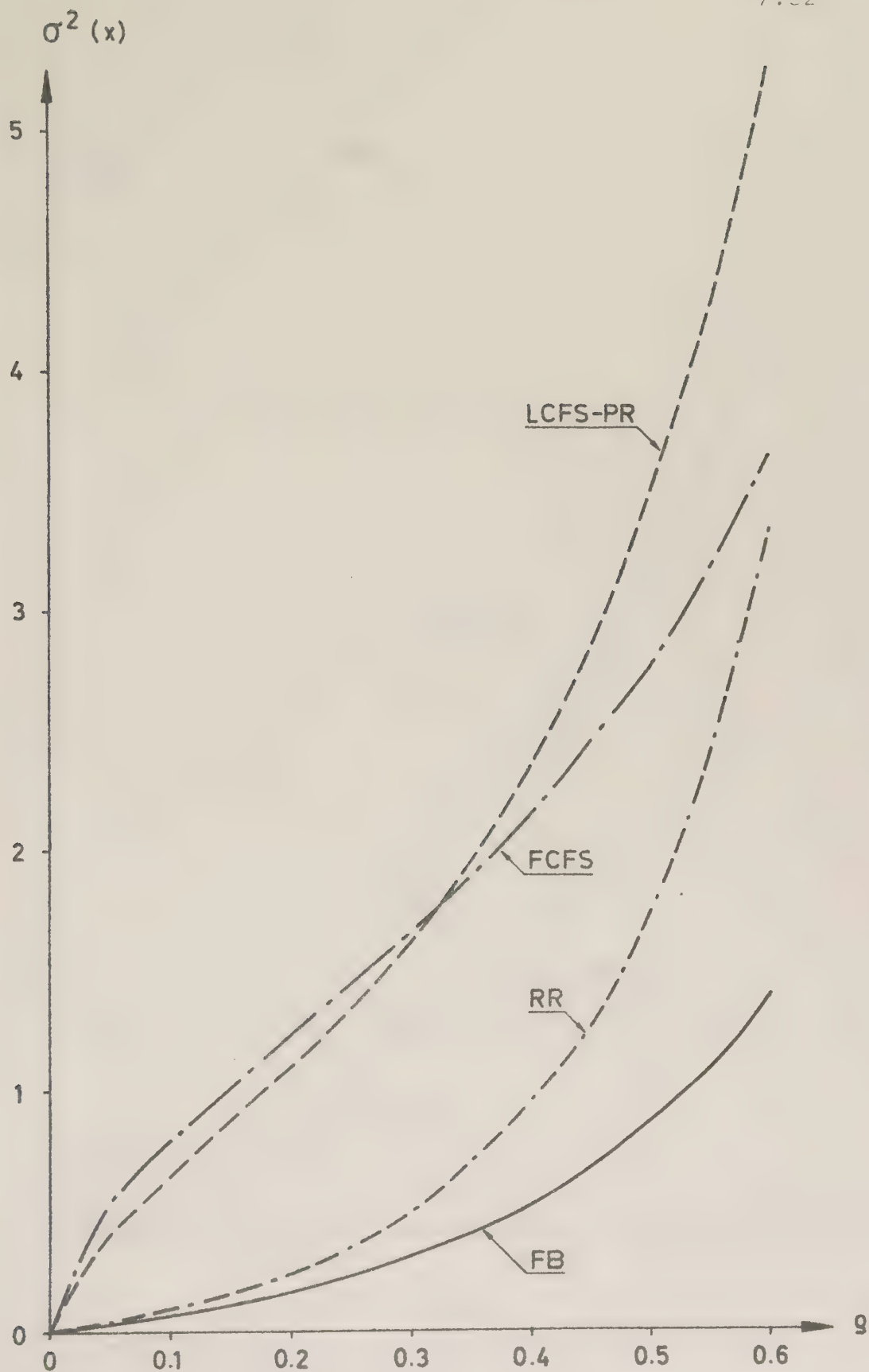


Fig. 7:4. Variance of conditional response time.
 $M/H_2/1$. $x=1$.

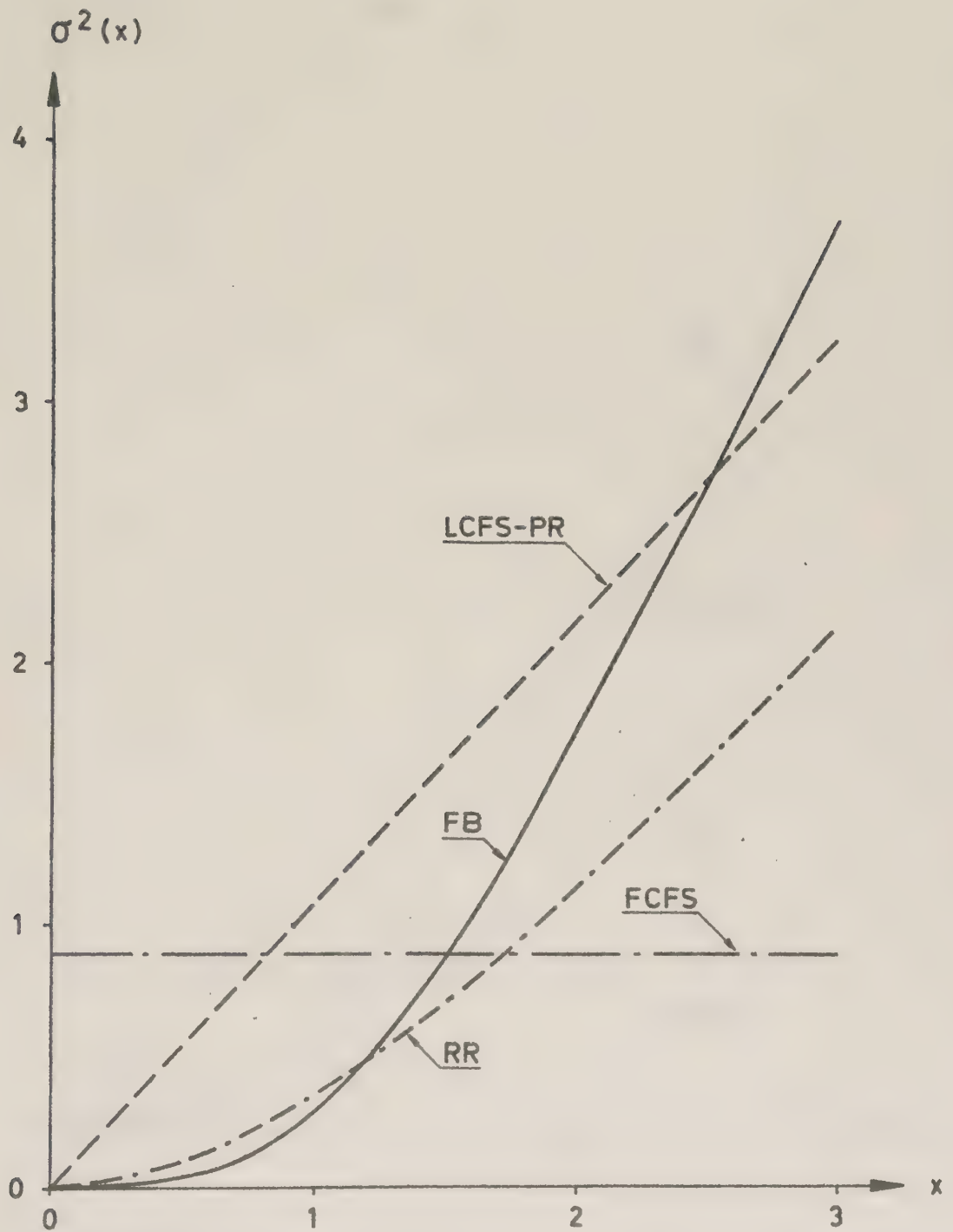


Fig. 7:5. Variance of conditional response time.
M/M/1. ρ constant.

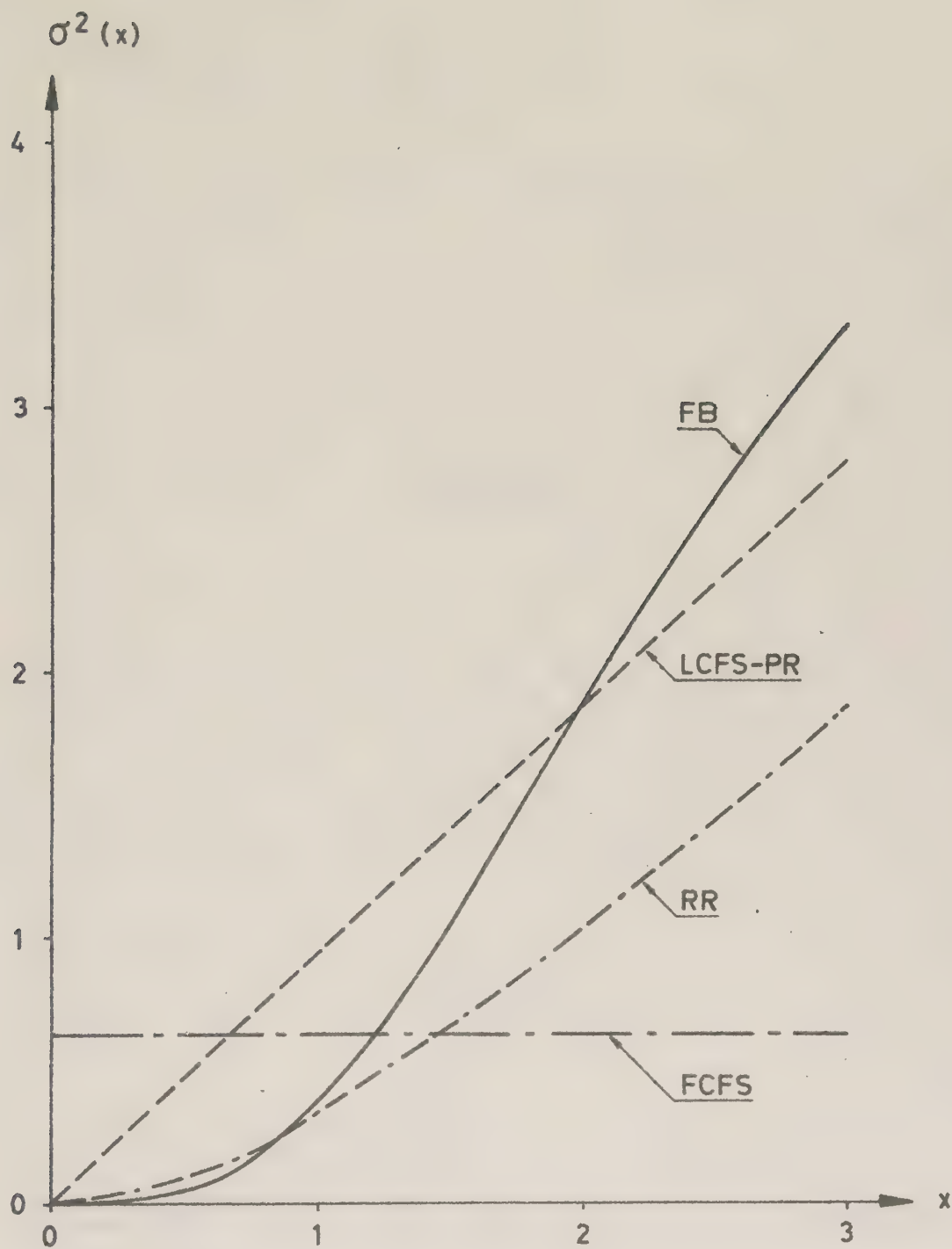


Fig. 7:6. Variance of conditional response time.
 $M/E_2/1$. ρ constant.

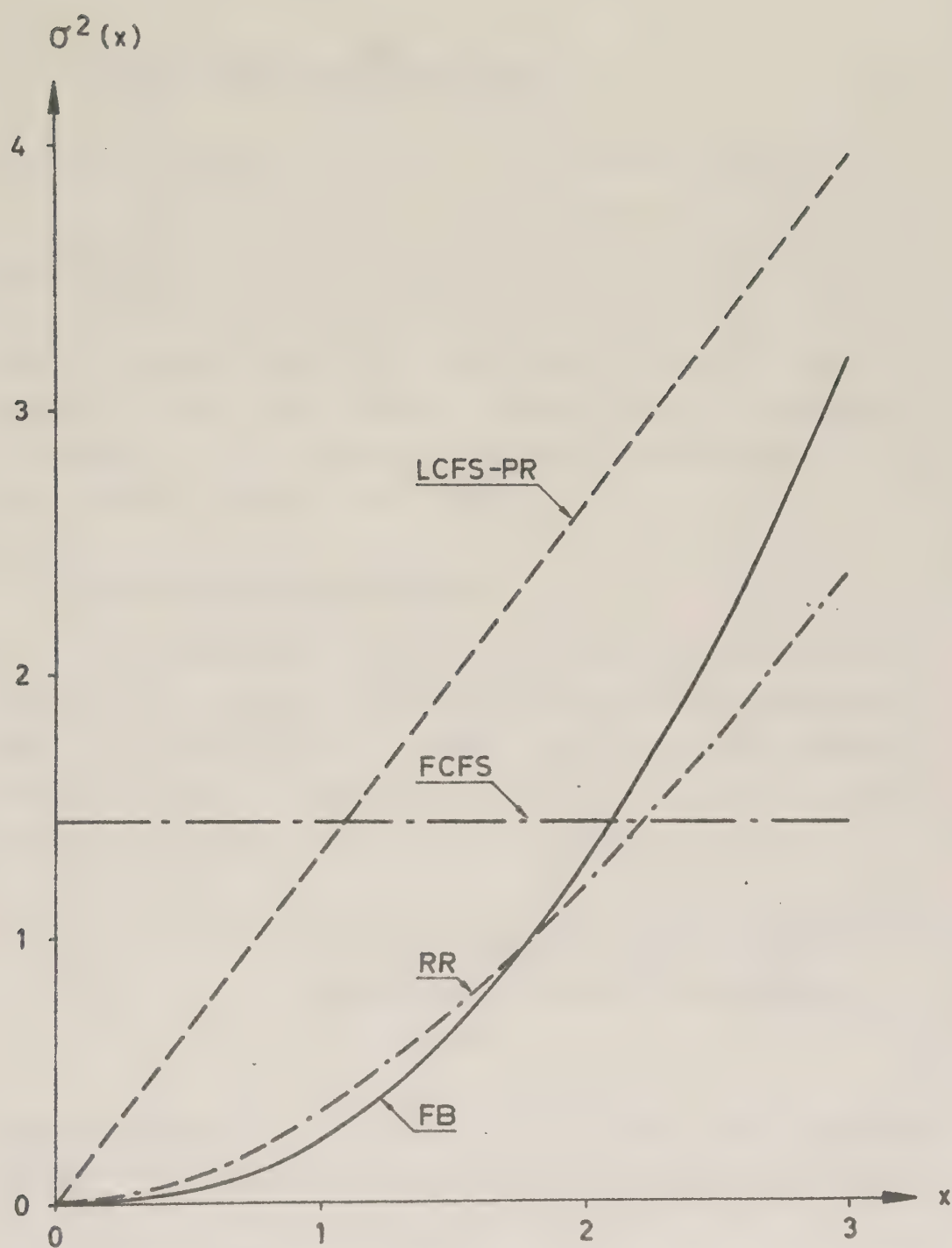


Fig. 7:7. Variance of conditional response time.
 $M/H_2/1$. ρ constant.

8. . Optimal Scheduling Algorithms

As stated in the introduction considerable effort has been put forth, during the last decade, in the analysis of given scheduling algorithms for time-sharing systems. However, little attention has been given to the important problems of defining some appropriate cost function and then of finding that algorithm which is optimal with respect to this cost function. We are now in a position to present a first attempt in facing this important optimization problem (KLEI 75c).

8.1 The Optimization Problem

Let us now consider a cost function for a time-sharing system. This we do by defining the following linear cost rate. Let $C(W,x)$ be the "system" cost rate (per average waiting second) of delaying a customer if he has a total service-requirement of x seconds. That is

$C(W,x)W(x)$ = Expected cost to the system for delaying
by $W(x)$ seconds (on the average) a
customer whose service requirement
is x seconds.

From the definition of $C(W,x)$, it is clear that the total average cost per customer, denoted by C may be defined as

$$C = \int_0^{\infty} C(W,x) W(x) b(x) dx \quad (8.1)$$

We look upon C as the average cost/customer that the system must pay. Our objective is to minimize C with respect to W , where we recognise that W is determined by the scheduling algorithm:, the service time distribution $B(x)$

also affects $W(x)$, but we assume $B(x)$ to be a given and not a function which we are permitted to vary in the optimization problem.

We know, e.g. from chapter 6, that the function $W(x)$ must obey at least 4 different constraints, namely:

- (i) $\frac{dW}{dx} \geq 0$
- (ii) $W \leq W_U$ (upper bound)
- (iii) $W \geq W_L$ (lower bound)
- (iv) $\int_0^{\infty} W(x)(1-B(x))dx = \text{constant (independent of the scheduling algorithm)} =$

$$= \frac{\overline{\rho x^2}}{2(1-\rho)} \triangleq L$$

where $W_U \geq W_L \geq 0$

The optimization problem may now be formulated as

$$\min_W C = \min_W \int_0^{\infty} C(W,x) W(x) b(x) dx \quad (8.2)$$

under the conditions (i) - (iv) (where $W(x)$ is varied by varying the scheduling algorithm). We recognize this problem as belonging to the class of calculus of variations problems. Unfortunately, it has been impossible to find a computationally efficient mathematical procedure which solves such a problem, excluding mathematical programming (which requires enormous amounts of computation). However, using just one of the conditions (the conservation law) it is possible to obtain a relatively simple expression for the W

that optimizes C . For simplicity we assume that W is a smooth function; unfortunately this means that the large and interesting class of multilevel systems (KLEI 70b, KLEI 72b), are not included among the algorithms we are looking for (however, the selfish scheduling algorithms (HSU 71, KLEI 70a, KLEI 75b) which may use these multilevel systems will smooth out their discontinuities and are included among our algorithms). Of greater importance is the observation that since we have kept just one of the four conditions, we may end up with a solution for $W(x)$ which violates some (or all) of the three other conditions.

Yet more annoying is the fact that even if we do find a W which falls into the class of functions defined by (i) - (iv), it still may not be feasible and further it may be that we have no idea how to implement such a scheduling algorithm even if it is feasible.

These problems arise since there may exist other constraints on W unknown to us at this time. That is, we can make use of some known necessary conditions on $W(x)$, but are currently unable to state the necessary and sufficient conditions on W . However, we make use of the following important observation:

If we optimize over a constraint space which includes some, but not necessarily all, of the constraints, then any solution which we obtain and which is also realizable by means of a known algorithm must be the true optimum solution (obviously it must satisfy all the constraints if it is realizable).

Thus the optimization problem may be formulated as

$$\min_W \int_0^{\infty} C(W,x) W(x) b(x) dx \quad (8.3)$$

Under the constraint

$$\int_0^{\infty} W(x) (1-B(x)) dx = L \quad (8.4)$$

A straightforward approach (using the Lagrange-multiplier technique) gives us that the optimal W is determined through the following expression (where k is a Lagrange multiplier):

$$\frac{\partial}{\partial W} \{C(W,x)W\} b(x) = k(1-B(x)) \quad (8.5)$$

As pointed out above, an optimal W , although possibly infeasible can be derived from this relation.

We therefore conclude that the result of this optimization procedure is a class of optimal W only some of which are feasible.

In order to proceed with some examples and some special cases, we require the specification of the cost function $C(W,x)$. It is difficult to find a generally agreed-upon function of this type, and so we are left in the position of having to invent some. This we do in the following sections as we study some examples and some extensions.

8.2 Simple Examples

In this section we demonstrate the above method through some simple, yet important, examples and also show that it is possible to end up in the infeasible region.

1) Let us choose

$$C(W, x) = W$$

$$b(x) = \mu e^{-\mu x}$$

This is a reasonable, but simplistic cost function.

From equation (8.5) we find that

$$W = \frac{k}{2\mu}$$

and therefore $W(x)$ is independent of x .

One scheduling algorithm which gives us a W independent of x is FCFS (First Come First Served) and so FCFS is one optimal choice (from among many, in fact from among all non-preemptive algorithms which operate independent of the service time).

2) Here we choose

$$C(W, x) = \frac{W}{x+a} \text{ where } a \geq 0, a \text{ constant}$$

$$b(x) = \mu e^{-\mu x}$$

This cost function is fairly reasonable in that it behaves sensibly with regard to W and x .

The optimal W must be (Eq. (8.5))

$$W = \frac{k}{2\mu} (x + a)$$

An algorithm which gives us this optimal W is one picked from the SSA-family namely SRR (Selfish Round Robin) (HSU 71, KLEI 70a, KLEI 75b)!

3) Now consider

$$C(W, x) = \frac{W}{x}$$

$$b(x) = 2\mu (2\mu x) e^{-2\mu x}$$

This combination once again gives us that SRR is optimal! On the other hand, with $b(x) = \mu e^{-\mu x}$ we find that the Processor-shared Round Robin (RR) algorithm (COFF 70, KLEI 64, KLEI 75b) is optimal.

4) If we let

$$C(W, x) = \frac{W}{x^2 + a}$$

$$b(x) = \mu e^{-\mu x}$$

We get that

$$W(x) = \frac{k}{2\mu} (x^2 + a)$$

Unfortunately this W violates the upper bound W_U , which requires that $W_U \sim \frac{\rho}{1-\rho} x$ when x is large and therefore the optimal W we have found is infeasible since it is proportional to x^2 .

5) Lastly consider

$$C(W,x) = W$$

$$b(x) = 2\mu (2\mu x) e^{-2\mu x}$$

This gives us a W of the form

$$W = \frac{k}{4\mu} \left(1 + \frac{1}{2\mu x}\right)$$

The solution is such that,

$\frac{dW}{dx} < 0$ and since this violates constraint (i) we see that this solution is also infeasible.

8.3 The Problem in Reverse (i.e. "Stacking" the Deck)

In the previous section we have shown that by altering some of the assumptions for $C(W,x)$ and $b(x)$, W could easily end up in the infeasible region. Sometimes we were "lucky", (and sometimes we were not!). We wish to improve our ability to find realizable solutions. This we do by judiciously choosing the right form for $C(W,x)$. Thus we shall in effect force the optimal W to belong to the class of W which we know how to implement. The trick is to "turn the problem around" and seek that class of cost functions for which a given feasible W is optimal (i.e., one known to be implementable). By so doing, we easily find the $C(W,x)$ which minimize C for a given feasible W instead of seeking the W which minimizes C for a given $C(W,x)$!

Let us restrict the cost function to be of the following form

$$C(W,x) = f(x) W^i \quad (8.6)$$

where $i \geq 1$. $f(x)$ is to be specified in advance and is totally independent of $W(x)$.

Let us assume, for example that $i = 1$; that is, $C(W,x) = f(x)W$.

From Eq. (8.5) we see that the optimal W will then be of the form:

$$W(x) = \frac{k}{2} \frac{1-B(x)}{f(x)b(x)} \quad (8.7)$$

Let us now demonstrate the technique outlined above for finding a suitable $f(x)$. We assume that we have a given scheduling algorithm and we wish to find the $C(W,x)$ which is minimized by this algorithm. (Throughout the examples

below, we restrict ourselves to the case where $i = 1$).

- 1) As a first example, assume that the scheduling algorithm is FCFS; then we know that (KLEI 75a)

$$W(x) = \frac{\lambda x^2}{2(1-\rho)} = \text{constant}^*$$

- (i) If $b(x) = \mu e^{-\mu x}$, then from Eq. (8.7) $f(x) = \text{constant}$

This is example 1, sect. 8.2

- (ii) If $b(x) = r \frac{(r\mu x)^{r-1}}{(r-1)!} e^{-r\mu x}$ (r-stage Erlang) then

$$f(x) = (\text{constant}) \frac{1 + r\mu x + \dots + \frac{(r\mu x)^{r-1}}{(r-1)!}}{\frac{(r\mu x)^{r-1}}{(r-1)!}}$$

- (iii) If $b(x) = \begin{cases} \mu/2 & 0 \leq x \leq 2/\mu \\ 0 & x > 2/\mu \end{cases}$

then

$$f(x) = \text{constant} \left(1 - \frac{2x}{\mu}\right)$$

- (iv) If $b(x) = \sum_{j=1}^r \alpha_j \mu_j e^{-\mu_j x}$ where $\sum_{j=1}^r \alpha_j = 1$

and $\alpha_j \geq 0$ (Hyper-exponential)

then

$$f(x) = (\text{constant}) \frac{\sum \alpha_j e^{-\mu_j x}}{\sum \alpha_j \mu_j e^{-\mu_j x}}$$

*The use of the word "constant" indicates some value independent of x . This value may differ in each of the equations below.

- 2) As a second example we assume that the algorithm is (processor-shared) Round-Robin, for which (KLEI 75b)

$$W(x) = \frac{\rho}{1-\rho} x$$

- (i) If $b(x) = \mu e^{-\mu x}$ then

$$f(x) = \frac{\text{constant}}{x}$$

This is example 2, sect. 8.2.

- (ii) If $b(x) = r\mu \frac{(r\mu x)^{r-1}}{(r-1)!} e^{-r\mu x}$ then

$$f(x) = (\text{constant}) \frac{1}{x} \frac{1 + r\mu x + \dots + \frac{(r\mu x)^{r-1}}{(r-1)!}}{(r\mu x)^{r-1}/(r-1)!}$$

- (iii) If $b(x) = \begin{cases} \mu/2 & 0 \leq x \leq 2/\mu \\ 0 & x > 2/\mu \end{cases}$

then

$$f(x) = (\text{constant}) \left(\frac{1}{x} - \frac{\mu}{2} \right)$$

- (iv) If $b(x) = \sum \alpha_j \mu_j e^{-\mu_j x}$

then

$$f(x) = (\text{constant}) \frac{\sum \alpha_j e^{-\mu_j x}}{\sum \alpha_j \mu_j x e^{-\mu_j x}}$$

- 3) Now assume that the algorithm is a member of the SSA-family, specifically SRR(Selfish Round-Robin), then we know

$$W(x) = (x+a)k_1$$

where k_1 and a are constants such that $k_1, a > 0$.

(i) If $b(x) = \mu e^{-\mu x}$ then

$$f(x) = \frac{\text{constant}}{x+a}$$

This is example 2, sect 8.2 again.

(ii) If $b(x) = r\mu \frac{(r\mu x)^{r-1}}{(r-1)!} e^{-r\mu x}$ then

$$f(x) = \frac{\text{constant}}{x+a} \frac{1 + r\mu x + \dots + \frac{(r\mu x)^{r-1}}{(r-1)!}}{\frac{(r\mu x)^{r-1}}{(r-1)!}}$$

(iii) If $b(x) = \begin{cases} \mu/2 & 0 \leq x \leq 2/\mu \\ 0 & x > 2/\mu \end{cases}$

then

$$f(x) = (\text{constant}) \left(\frac{1 - \frac{\mu x}{2}}{x+a} \right)$$

Thus we see that some well-known algorithms are optimal with respect to some (possibly unusual) cost functions. It is interesting to note that the SRR algorithm is optimal for $C(W, x) = \frac{W}{x+a}$ when service time is exponentially distributed. We repeat that this cost function is quite a reasonable and simple one.

In the next chapter we will demonstrate through some numerical examples the applicability of the optimization.

9. Numerical Results for the Optimal Theory

In the previous chapter a theory was presented describing how to find optimal scheduling algorithms. In section 8.3 we studied the problem in reverse or, as we called it, "stacking the deck". Our trick in that section was to turn the problem around and look for the class of cost-functions for which a given feasible $W(x)$ was optimal. A number of simple examples were also given and formulae for the function $f(x)$ presented. In this chapter we will demonstrate through a few examples how the service time distribution affects the optimal cost. In particular we will determine the function $f(x)$ for a given density function $b_0(x)$ and then compute the corresponding cost for a different density function $b(x)$. Obviously this computation is a very essential one, since the service time distribution may not in real life situations, be known to us.

We could easily produce an almost infinite number of examples. We feel, however, that it will be appropriate to limit the number of examples chosen and therefore we will concentrate on costfunctions that are of the form

$$C(W,x) = W(x) f(x) \qquad (9.1)$$

Furthermore we will use the Erlang-r and H_2 distribution for the service times. We will in fact use a special type of the H_2 - distribution, namely

$$b(x) = 2\alpha^2 e^{-2\alpha x} + 2(1-\alpha)^2 e^{-2(1-\alpha)x} \quad (9.2)$$

where

$$0 \leq \alpha \leq 1$$

We note that

$$\overline{x} = 1 \quad (9.3)$$

$$\overline{x^2} = \frac{1}{2\alpha(1-\alpha)} \quad (9.4)$$

which gives us an opportunity to vary the coefficient of variation over a rather large interval.

Let us now demonstrate the influence on the cost when we vary $b(x)$ for some different scheduling algorithms. Our first example is one where the algorithm chosen is SRR. In figure 9:1 we show how the ratio cost divided by optimal cost vary when the service time distribution

is Erlang- r and the load ρ is held constant. r_{OPT} is the value for which we have computed $f(x)$. We can clearly see that if r - the number of stages, is less than r_{OPT} , the encountered cost is greater than the optimal cost. Intuitively this is the behaviour we expect, since the cost should be greater than the optimal cost. On the other hand we get a bit surprised when we see that the cost is actually less than the optimal cost when r is greater than r_{OPT} . But if we use this kind of intuition, we are on the wrong track since we will then expect the computed $f(x)$ to generate that cost function which is optimized by SRR when the service-time distribution is Erlang- r .

Of course this is not necessarily true, since that cost-function may yield quite another - possibly unfeasible - optimal algorithm when $b(x)$ is equal to an Erlang- r density function. We should instead emphasize what figure 9:1 shows, namely, the effect on the cost when $b(x)$ is varied. In reality, figure 9:1 gives us an indication of the error we do when we optimize for a certain r_{OPT} and the actual r is different. In figure 9:2, we depict as a second example the same quantity as in fig. 9:1. Once again the algorithm chosen is SRR, but this time the

service time distribution is the H_2 - distribution, as defined by (9.2). The curves must obviously be symmetrical, due to the special form of $b(x)$.

Although we could present many more, we have, as a last example, chosen the FB algorithm, figure 9:3, and - as in our first example - we depict the ratio cost divided by optimal cost as a function of r when the service time distribution is Erlang- r .

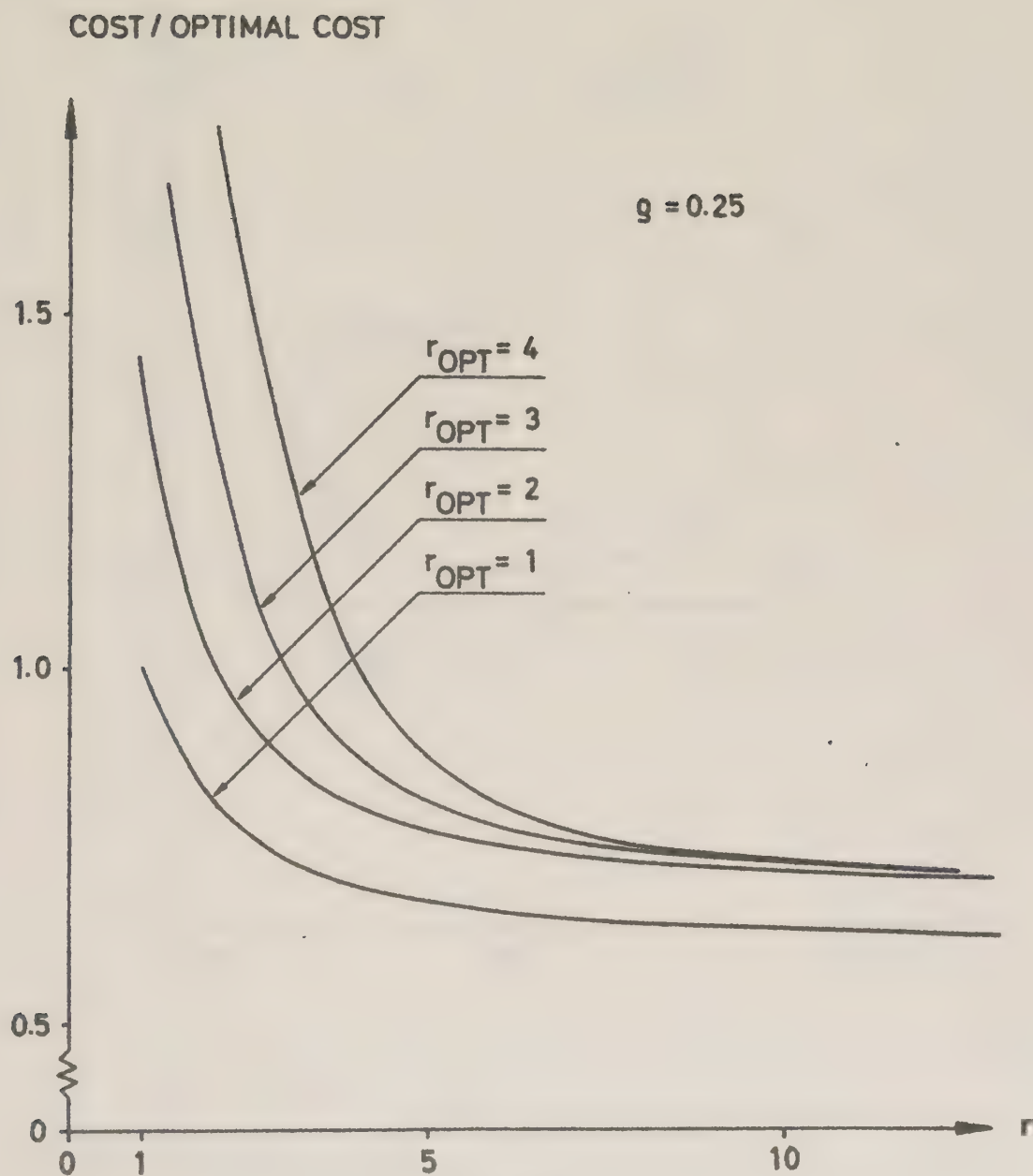


Fig. 9:1. Cost divided by optimal cost for an SRR system. Servicetimedistribution Erlang - r . ρ constant.

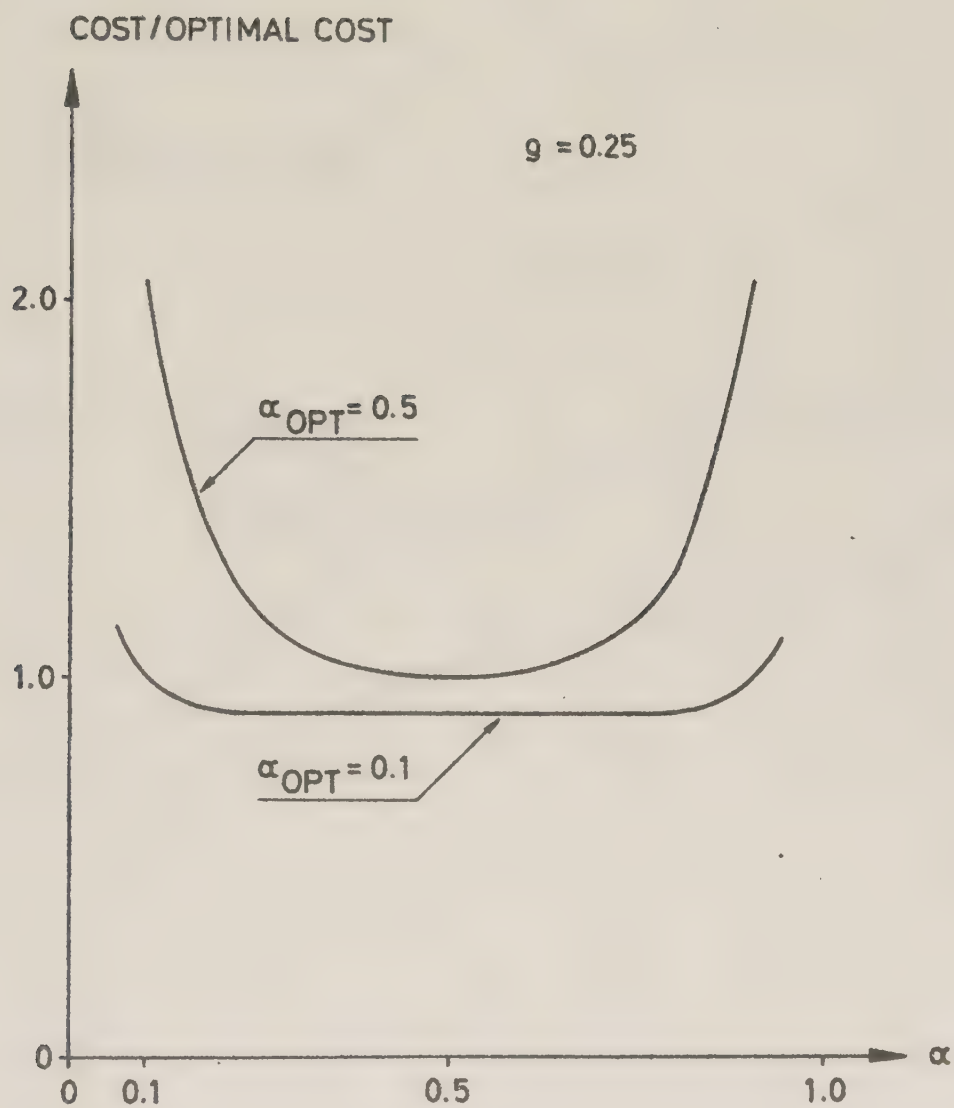


Fig. 9:2. Cost divided by optimal cost for an SRR system. Service time distribution H_2 . ρ constant.

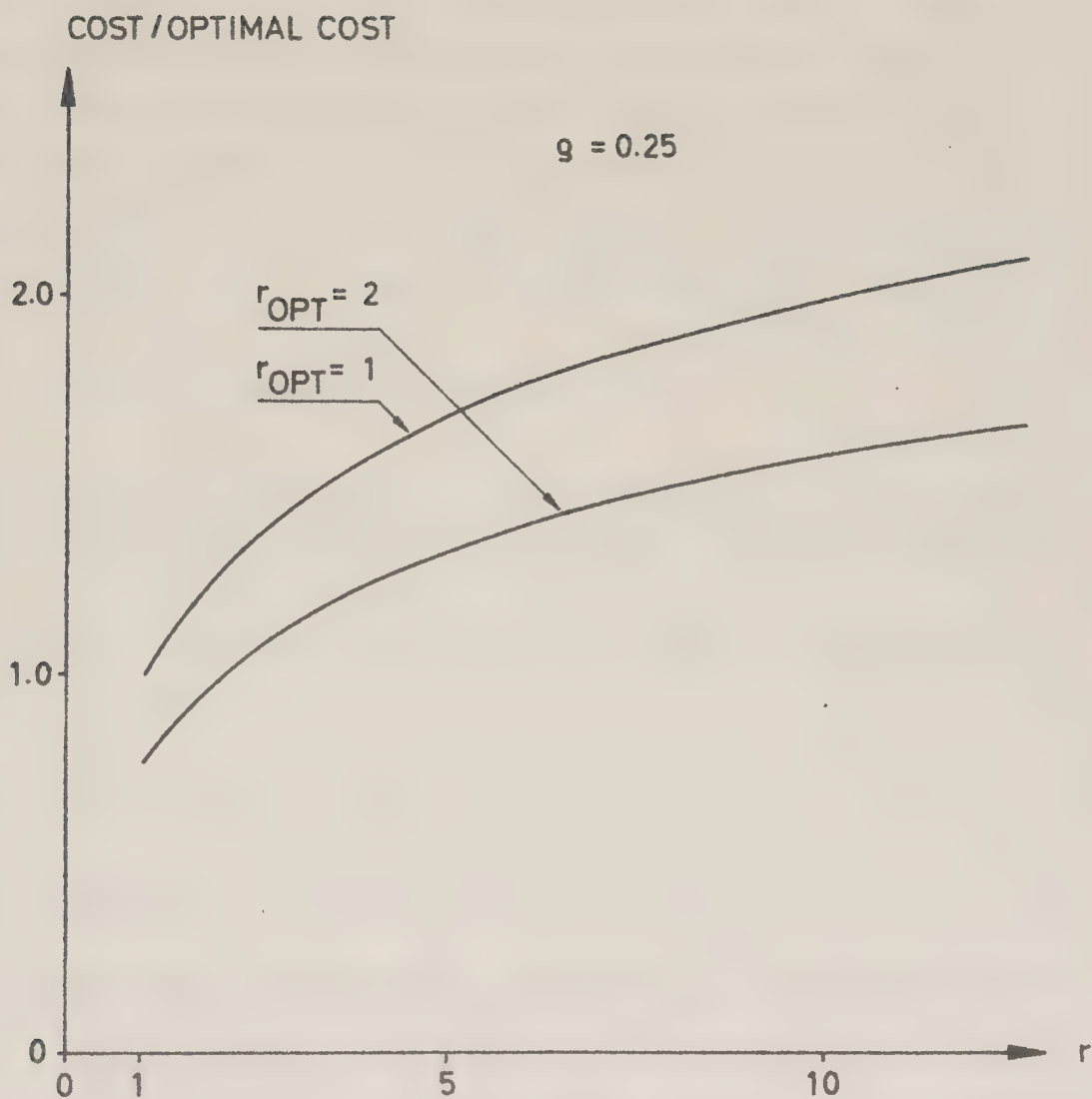


Fig. 9:3. Cost divided by optimal cost for an SFB system. Servicetimedistribution Erlang - r . ρ constant.

10. Attempts to Optimize Multi-Level System

In Chapter 8 we remarked that the optimization theory that was developed, did not include the multilevel systems. However, this annoying fact was reduced a bit since we also remarked that the class of selfish scheduling algorithms (SSA) could smooth out the discontinuities and thus make the multilevel system treatable by the general theory. In this chapter we are going to formulate two approaches to the optimal scheduling problem in a multilevel system. We use the same kind of cost function as in chapter 8. This means that we want to find the scheduling disciplines that minimize the cost C , where C is given through

$$C = \int_0^{\infty} C(W,x) W(x) b(x) dx \quad (10.1)$$

Since we are dealing with a multilevel system, what we want to do is find the optimal D_i 's and a_i 's, where $i = 1, 2, \dots$

Our first approach is to assume that the a_i 's are known to us. That is

$$a_1, a_2, \dots, a_N$$

are known and as usual $0 < a_1 < a_2 < \dots < a_N < a_{N+1} = \infty$

The optimization problem then reduces to, find the minimum of

$$C = \sum_{i=1}^{N+1} \int_{a_{i-1}}^{a_i} C(W,x) W(x) b(x) dx \quad (10.2)$$

with respect to the scheduling algorithms D_i . In chapter 4 we solved for $W(x)$ when D_i was one of FCFS, FB, RR and LCFS-PR. We now formulate the optimization as: "find the

D_i 's belonging to the class of the four scheduling algorithms that minimize the cost C ."

A computer program has been written that performs this optimization and in tables 10:1,2 we show some results for different cost functions.

The second approach is to assume that only N is given, the number of levels and then find the sequences $\{a_i\}_1^N$ and $\{D_i\}_1^N$ possibly by the aid of mathematical programming. In order to make the problem more tractable we have to fix a_N , the highermost breakpoint. This can be done f.i. by claiming that the probability that a service requirement is greater than a_N is kept below a certain fixed small number. The optimization problem can then be written as

$$\min \int_0^{a_N} C(W,x) W(x) b(x) dx \quad (10.3)$$

Let us write the integrand as

$$f(W,x) = C(W,x) W(x) b(x) \quad (10.4)$$

where we have to remember that it is only $W(x)$ that we may vary.

Hence, the problem is written as

$$\min_{W, \{a_i\}_1^{N-1}} \int_0^{a_N} f(W,x) dx \quad (10.5)$$

We now use a dynamic programming method and introduce

$$\min_{W,x} \left\{ \int_0^x f(W,y) dy + \int_x^{a_N} f(W,y) dy \right\} \quad (10.6)$$

Furthermore, let

$$C_i(a_{i-1}, a_i) = \text{the cost for the } i^{\text{th}} \text{ level} \quad (10.7)$$

then

$$\min_{\{x_i\}} \sum_{i=1}^N C_i(a_{i-1}, a_i) = \min_{\substack{\sum_{i=1}^N x_i = a_N \\ 1}} \sum_{i=1}^N C_i(x_i) \quad (10.8)$$

where $x_i = a_i - a_{i-1}$

Define

$$f_1(x) = \min_{0 \leq x_1 \leq x} C_1(x_1) \quad (10.9)$$

then according to the principle of dynamic programming

$$f_k(x) = \min_{0 \leq x_k \leq x} \{C_k(x_k) + f_{k-1}(x - x_k)\} \quad (10.10)$$

for $k \geq 1$

Note that the optimization has also to be performed over the four permitted disciplines. Once again a computer seems to be most appropriate, since the problem is too complex for manipulation by hand. However, even a computer may be insufficient since the computational complexity is vast. No attempt has been made to implement a computer program that would perform the optimization. We feel that the results eventually achieved would not greatly enhance our ability to build optimal TS-systems.

Level	FCFS	FB	RR	LCFS-PR	OPTIMAL
$a_1 = 1$	0.024	0.050	0.034	0.034	FCFS
$a_2 - a_1 = 1$	0.473	0.523	0.511	0.480	FCFS
$a_3 - a_2 = 1$	0.872	0.889	0.887	0.874	FCFS
$\geq a_3$	1.560	1.592	1.589	1.569	FCFS

Table 10:1. Computed cost for an $N = 3$ level system,

$a_i = i$, $i = 1, 2, 3$. Costfunction $C(W(x), x) = W(x)$

M/M/1 system with parameters $\lambda = 0.5$ and $\bar{x} = 1$.

Level	FCFS	FB	RR	LCFS-PR	OPTIMAL
$a_1 = 1$	0.028	0.039	0.029	0.029	FCFS
$a_2 - a_1 = 1$	0.245	0.253	0.250	0.244	LCFS-PR
$a_3 - a_2 = 1$	0.298	0.298	0.298	0.297	LCFS-PR
$\geq a_3$	0.339	0.338	0.338	0.338	FB

Table 10:2 Computed cost for an $N = 3$ level system,

$a_i = i$, $i = 1, 2, 3$. Costfunction $C(W(x), x) = W(x)/(x + \bar{x}/2)$

M/M/1 system with parameters $\lambda = 0.5$ and $\bar{x} = 1$.

11. Conclusions and Suggestions for Future Research

We have in this dissertation produced pertinent analytical results for the mean and variance of conditional response time for a large class of scheduling algorithms. Furthermore, a theory for the optimization of time-shared systems has been introduced. The results obtained especially those in chapters 4, 7 and 8 certainly do enhance our understanding of the complex operation of a time-sharing computer system.

It is, however, natural to ask what kind of extensions and future research needs to be done in this area. In chapter 6 we presented some necessary conditions that a given response function has to follow. But that is only part of the answer; the question as to what are the necessary and sufficient conditions for a given response function to be feasible is not yet solved.

The results concerning the variance of conditional response time are certainly most encouraging. We feel, however, that it would be nice to obtain the solution for general service time distributions when the scheduling discipline is RR. The optimization theory as presented in chapter 8 needs further development. Obviously we would like to incorporate the other known constraints on the

response time in the optimization procedure. Note, however, that even if we manage to perform this task we can still end up in an unfeasible region since we do not know, at this time, all the necessary and sufficient conditions that a response function has to obey. Furthermore, the costfunction we used for the optimization may not be the best one. Perhaps the costfunction should also include the variance of conditional response time, or some other measure of the variability of the response time.

The assumption that the input process to the system is Poissonian, i.e. an infinite user-population, is obviously unrealistic. Thus the analysis of time - sharing systems when the user population is limited should be performed. We know that some research has been done in this area but to our knowledge the success has been limited.

REFERENCES

- COFF 66 Coffman, E.G., "Stochastic Models of Multiple and Time - Shared Computer Operations," Report No. 66 - 38, Dept. of Engineering, University of California at Los Angeles (June 1966).
- COFF 68 Coffman, E.G., Jr., and Kleinrock L., "Feedback Queueing Models for Time-Shared Systems," JACM" (October 1968) 15:549-576.
- COFF 70 Coffman, E.G., Jr., Muntz, R. R., and Trotter, H., "Waiting Time Distribution for Processor-Sharing Systems," JACM (January 1970) 17:123-130.
- COHE 69 Cohen, J.W., The Single Server Queue, Wiley (New York) 1969.
- COX 55 Cox, D.R., "A Use of Complex Probabilities in the Theory of Stochastic Processes," Proceedings Cambridge Philosophical Society (1955) 51:313-319.
- ERDE 54 Erdelyi, A., Tables of Integral Transforms, M^CGraw - Hill 1954.

- ESTR 67 Estrin, G., and Kleinrock, L., "Measures, Models and Measurements for Time-Shared Computer Utilities," Proc. of 22nd National Conference of the Association for Computing Machinery, sponsored by the Association for Computing Machinery, Aug. 29-31, 1967, Washington, D.C., (1967) pp. 85-96.
- FELL 66 Feller, W., Probability Theory and its Application Vol. II, Wiley (New York), 1966.
- HSU 71 Hsu, J., "Analysis of a Continuum of Processor-Sharing Models for Time-Shared Computer Systems," UCLA-ENG-7166, School of Engineering and Applied Science, Univ. of California, Los Angeles (October 1971)
- KLEI 64 Kleinrock, L., "Analysis of a Time-Shared Processor," Naval Research Logistics Quarterly, (March 1964) 11:59-73.
- KLEI 67 Kleinrock, L., "Time-Shared Systems: A Theoretical Treatment," JACM, (1967) 14:242-261.

- KLEI 70a Kleinrock, L., "A Continuum of Time-Sharing Scheduling Algorithms," AFIPS Conference Proc., Spring Joint Computer Conference (May 1970) pp. 453-458.
- KLEI 70b Kleinrock, L., and Muntz, R.R., "Multilevel Processor-Sharing Queueing Models for Time-Shared Models," Proc. Sixth International Teletraffic Congress, (August 1970) 341/1-341/8.
- KLEI 71a Kleinrock, L., Muntz, R.R., and Hsu, J., "Tight Bounds on Average Response Time for Processor-Sharing Models of Time-Shared Computer Systems," Proceedings of the International Federation for Information Processing Congress 71, Aug. 23-28, Ljubljana, Yugoslavia, North Holland Publishing Co. Amsterdam, 1971, TA-2, p. 124-133.
- KLEI 71b Kleinrock, L., Muntz, R.R., and Rodemich, E., "The Processor - Sharing Queueing Model for Time - Shared Systems with Bulk Arrivals," Networks Journal, (1971) 1:1-13.

- KLEI 72a Kleinrock, L., "A Selected Menu of Analytical Results for Time-Shared Computer Systems," in Systemprogrammierung, R. Oldenburg Verlag, Munich, Germany, (1972) pp. 45-73.
- KLEI 72b Kleinrock, L., and Muntz, R.R., "Processor-Sharing Queueing Models of Mixed Scheduling Disciplines for Time-Shared Systems," JACM, (July 1972), 19:464-482.
- KLEI 75a Kleinrock, L., Queueing Systems, Volume I: Theory, John Wiley and Sons, New York, 1975.
- KLEI 75b Kleinrock, L., Queueing Systems, Volume II: Computer Applications, John Wiley and Sons, New York, 1975.
- KLEI 75c Kleinrock, L., and Nilsson, A., "On Optimal Scheduling Algorithms for Time-Shared Systems," (To be published).
- LAVE 75 Lavenberg, S.S., and Shedler, G.S., "Derivation of Confidence Intervals for Work Rate Estimators in a Closed Queueing Network," SIAM J. Comput. (June 1975) Vol. 4, No.2. pp 108-124.

- MCKI 69 McKinney, J.M., "A Survey of Analytical Time-Sharing Models," Computing Surveys, (June 1969) 1: 105-116.
- MUNT 73 Muntz, R.R., "Network of Queues Models: Application to Computer System Modeling" UCLA - course notes.
- ODON 74 O'Donovan, T.M., "Distribution of Attained and Residual Service in General Queueing Systems," Operations Research (1974) 22: 57Q-575.
- RIOR 62 Riordan, J., Stochastic Service Systems, Wiley (New York) 1962.
- ROSS 70 Ross, S. H., Applied Probability Models with Optimization Applications, Holden-Day (San Francisco), 1970.
- SAKA 71 Sakata, M., Noguchi, S., and Dizumi, J., "An Analysis of the M/G/1 Queue under Round Robin Scheduling," Operations Research (1971) 19: 371-385.

SCHR 67

Schrage, L.E., "The Queue M/G/1 with Feedback to Lower Priority Queues," Management Science (1967) 13: 466-471.

TAKA 62

Takács, L., Introduction to the Theory of Queues, Oxford University Press (New York) 1962.

